

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN I

—o0o—



BÁO CÁO BÀI TẬP LỚN 1
NGÔN NGỮ LẬP TRÌNH PYTHON

Giảng viên hướng dẫn:

Sinh viên:

Mã sinh viên:

Lớp:

Niên khóa:

Hệ đào tạo:

Kim Ngọc Bách

Phùng Thu Hương

B23DCVT201

D23CQCEO6-B

2023 - 2028

Đại học chính quy

Hà Nội, 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN I

—o0o—



BÁO CÁO BÀI TẬP LỚN 1
NGÔN NGỮ LẬP TRÌNH PYTHON

Giảng viên hướng dẫn:

Sinh viên:

Mã sinh viên:

Lớp:

Niên khóa:

Hệ đào tạo:

Kim Ngọc Bách

Phùng Thu Hương

B23DCVT201

D23CQCEO6-B

2023 - 2028

Đại học chính quy

Hà Nội, 2025

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm: (Bằng chữ:)

Hà Nội, ngày tháng năm 20...

Giảng viên

Mục lục

1	Thu thập dữ liệu cầu thủ từ fbref.com	6
1.1	Lý do lựa chọn thư viện Selenium	6
1.2	Thư viện sử dụng	7
1.3	Quy trình cào dữ liệu (chi tiết theo mã nguồn)	7
1.3.1	Bước 1: Đọc cấu hình bảng cần lấy dữ liệu	7
1.3.2	Bước 2: Tải trang bằng Selenium	7
1.3.3	Bước 3: Phân tích HTML bằng BeautifulSoup	8
1.3.4	Bước 4: Gộp dữ liệu từ nhiều bảng	8
1.3.5	Bước 5: Lọc cầu thủ đủ điều kiện (> 90 phút)	8
1.3.6	Bước 6: Sắp xếp theo tên	8
1.3.7	Bước 7: Chuẩn hóa cột dữ liệu	8
1.3.8	Bước 8: Xuất dữ liệu ra tệp CSV	9
1.4	Kết luận	9
2	Phân tích và trực quan hóa dữ liệu	10
2.1	Phân tích top 3 cầu thủ cao nhất và thấp nhất theo từng chỉ số thống kê .	10
2.1.1	Khái quát về yêu cầu của bài toán	10
2.1.2	Khái quát logic code chính (top_3.py)	10
2.1.3	Xử lý 'N/a' và các giá trị không hợp lệ	10
2.1.4	Các bước thực thi	11
2.1.5	Kết quả (top_3.txt)	11
2.2	Tính toán thống kê mô tả (median, mean, std dev) cho dữ liệu cầu thủ .	12
2.2.1	Giới thiệu	12
2.2.2	Lựa chọn thư viện	13
2.2.3	Logic và quy trình thực hiện (calculating_statistics.py)	13
2.2.4	Kết quả	14
2.3	Trực quan hóa phân phối thống kê cầu thủ bằng biểu đồ histogram . . .	15
2.3.1	Mục tiêu	15
2.3.2	Lựa chọn Thư viện	15
2.3.3	Các bước thực hiện	16
2.3.4	Hạn chế của việc sử dụng biểu đồ Histogram	17
2.4	Phân tích hiệu suất các đội bóng Premier League 2024–2025	18
2.4.1	Giới thiệu	18
2.4.2	Ý tưởng chính	18
2.4.3	Quy trình thực thi	18
2.4.4	Kết quả	20

3	Phân cụm Cầu thủ bằng K-Means và PCA	22
3.1	Giới thiệu và chuẩn bị dữ liệu	22
3.1.1	Giới thiệu	22
3.1.2	Chuẩn bị dữ liệu	22
3.2	Phân cụm và giảm chiều dữ liệu	23
3.2.1	Xác định số lượng cụm k tối ưu	23
3.2.2	Giảm chiều dữ liệu với PCA	24
3.2.3	Trực quan hóa các cụm	24
3.2.4	Phân tích cụm	24
3.3	Kết luận	26
4	Ước tính giá trị cầu thủ	27
4.1	Giới thiệu và cách thu thập dữ liệu	27
4.1.1	Giới thiệu	27
4.1.2	Thu thập dữ liệu	27
4.2	Phương pháp ước tính giá trị cầu thủ	27
4.2.1	Lựa chọn đặc trưng (Feature Selection)	27
4.2.2	Lựa chọn mô hình	28
4.2.3	Phân tích hình ảnh trực quan hóa về hiệu suất của mô hình	28
4.2.4	Huấn luyện và đánh giá mô hình	31
4.2.5	Kết quả	31
4.2.6	Hạn chế	32
4.3	Kết luận và kiến nghị	32

Danh sách hình vẽ

1.1	Hình ảnh minh họa file results.csv	9
2.1	Hình ảnh kết quả file top_3.txt	12
2.2	Hình ảnh 1 phần của file results2.csv	15
2.3	Hình ảnh ví dụ kết quả phân tích chi tiết cho từng chỉ số	20
2.4	Hình ảnh ví dụ về bảng xếp hạng tổng hợp các đội	20
3.1	Hình ảnh biểu đồ Elbow	23
3.2	Hình ảnh biểu đồ phân tán 2D	26
4.1	Hình ảnh file results.csv	28
4.2	Hình ảnh của biểu đồ feature_importance	29
4.3	Hình ảnh của biểu đồ pred_vs_actual	30
4.4	Hình ảnh của biểu đồ residuals	31
4.5	Hình ảnh kết quả đánh giá mô hình	31

Mở đầu

Báo cáo này được thực hiện nhằm đáp ứng yêu cầu của bài tập lớn số 1 trong khuôn khổ môn học Lập trình Python. Trọng tâm của bài tập là áp dụng các kỹ thuật thu thập, phân tích và mô hình hóa dữ liệu để làm sáng tỏ thông tin về màn trình diễn của các cầu thủ tại giải bóng đá Ngoại hạng Anh mùa giải 2024-2025.

Nguồn dữ liệu chính được sử dụng trong nghiên cứu này là trang web fbref.com, nơi cung cấp các số liệu thống kê chi tiết về hiệu suất thi đấu của cầu thủ. Phạm vi thu thập dữ liệu bao gồm tất cả các cầu thủ đã có thời gian ra sân trên 90 phút tại giải Ngoại hạng Anh mùa giải 2024-2025. Bên cạnh đó, dữ liệu về giá trị chuyển nhượng ước tính của các cầu thủ cũng được thu thập từ trang web footballtransfers.com đối với các cầu thủ đã thi đấu trên 900 phút. Toàn bộ quá trình thu thập dữ liệu được thực hiện vào ngày 2 tháng 5 năm 2025. Do đó, mọi phân tích và kết quả được trình bày trong báo cáo này phản ánh tình hình và số liệu của các cầu thủ tính đến thời điểm cụ thể này của mùa giải.

Nội dung của báo cáo được tổ chức thành bốn chương chính, cụ thể như sau:

Chương 1. Thu thập dữ liệu cầu thủ từ fbref.com: Chương này trình bày chi tiết quy trình thu thập dữ liệu thống kê cầu thủ từ trang web fbref.com bằng cách sử dụng các công cụ tự động hóa như Selenium và BeautifulSoup. Các bước làm sạch và lưu trữ dữ liệu ban đầu cũng được mô tả cụ thể. Chương này đáp ứng các yêu cầu của phần I của bài tập.

Chương 2. Phân tích và Trực quan hóa Dữ liệu: Chương này tập trung vào việc phân tích mô tả dữ liệu thu thập được. Các phân tích bao gồm xác định top 3 cầu thủ có thành tích cao nhất và bottom 3 cầu thủ có thành tích thấp nhất cho từng chỉ số, tính toán các giá trị thống kê tóm tắt (trung bình, trung vị, độ lệch chuẩn) cho từng chỉ số trên toàn giải đấu và cho từng đội bóng. Ngoài ra, chương này cũng trình bày các biểu đồ histogram minh họa sự phân phối của dữ liệu và xác định đội bóng có hiệu suất tốt nhất dựa trên các chỉ số thống kê.

Chương 3. Phân cụm Cầu thủ bằng K-Means và PCA: Chương này áp dụng thuật toán phân cụm K-Means để nhóm các cầu thủ thành các cụm dựa trên các đặc điểm thống kê tương đồng. Kỹ thuật giảm chiều dữ liệu PCA (Phân tích Thành phần Chính) được sử dụng để trực quan hóa các cụm trên biểu đồ hai chiều, từ đó làm rõ hơn sự khác biệt giữa các nhóm cầu thủ.

Chương 4. Ước tính giá trị cầu thủ: Chương này tập trung vào việc thu thập dữ liệu giá trị chuyển nhượng ước tính của các cầu thủ từ trang web footballtransfers.com và đề xuất một phương pháp ước tính giá trị thị trường của họ. Phương pháp này dựa trên việc xây dựng mô hình học máy (Gradient Boosting Regressor) với các đặc trưng được lựa chọn từ dữ liệu thống kê hiệu suất (thu thập ở Chương 1), thông tin cơ bản của cầu thủ và giá trị chuyển nhượng lịch sử. Quá trình lựa chọn đặc trưng, huấn luyện và đánh giá mô hình được trình bày chi tiết.

Chương 1

Thu thập dữ liệu cầu thủ từ fbref.com

Phần này em trình bày chương trình Python để thu thập tự động dữ liệu thống kê của các cầu thủ thi đấu tại giải Ngoại hạng Anh (English Premier League - EPL) mùa 2024-2025 từ trang web <https://fbref.com>. Các chỉ số cần thu thập bao gồm nhiều khía cạnh: quốc tịch, tuổi, thời gian thi đấu, phong độ thi đấu, chuyền bóng, sút, thủ môn, v.v.

Chương trình được tổ chức thành các file Python module:

- `football_stats_scraper.py`: mã nguồn chính gồm toàn bộ quy trình tự động tải, phân tích và lưu dữ liệu.
- `config.py`: lưu các cấu hình như URL cơ sở, định danh bảng (ID HTML), danh sách chỉ số cần thu thập (fields), các cột cần xuất hiện trong kết quả (CSV_COLUMNS), đường dẫn tệp kết quả (CSV_FILE), v.v.

Sau khi chạy, chương trình sẽ tạo ra một tệp kết quả có tên:

- `results.csv`: chứa các cầu thủ có tổng số phút thi đấu > 90, đã được lọc và chuẩn hóa, sẵn sàng dùng cho các phân tích tiếp theo.

Tệp kết quả được lưu tại: `Report/OUTPUT_BAI1/results.csv`

1.1 Lý do lựa chọn thư viện Selenium

Trang web fbref.com là một trang web hiện đại, trong đó các bảng thống kê không được tải sẵn dưới dạng HTML tĩnh mà được sinh ra thông qua JavaScript sau khi trang đã hiển thị. Điều này khiến các thư viện truyền thống như `requests` hoặc `urllib` không thể lấy được dữ liệu hoàn chỉnh do chúng chỉ đọc mã HTML ban đầu.

Trong trường hợp này, Selenium là lựa chọn phù hợp nhất vì các lý do sau:

- **Render JavaScript hoàn chỉnh:** Selenium điều khiển trình duyệt thực tế (như Chrome), do đó có thể thực hiện toàn bộ quá trình tải trang, bao gồm cả các đoạn mã JavaScript sinh ra bảng thống kê.
- **Tương tác động với DOM:** Selenium có thể chờ (`wait`) các phần tử cụ thể xuất hiện trước khi lấy dữ liệu. Điều này rất cần thiết vì bảng thống kê của fbref có thể mất vài giây để hiển thị sau khi vào trang.

- **Hỗ trợ mở rộng:** Nếu trong tương lai cần nhấn nút, chọn mùa giải khác, hoặc chuyển trang thì Selenium có thể xử lý dễ dàng nhờ mô phỏng hành vi người dùng.

Tóm lại, Selenium là công cụ thích hợp và mạnh mẽ để thu thập dữ liệu từ các trang web động như fbref.com. Khi kết hợp với BeautifulSoup, nó cung cấp cả khả năng tự động hóa lẫn khả năng phân tích HTML dễ dàng.

1.2 Thư viện sử dụng

- **selenium:** điều khiển trình duyệt Chrome tự động truy cập và tải các bảng dữ liệu có sử dụng JavaScript.
- **BeautifulSoup4 (bs4)** : phân tích HTML DOM sau khi trang đã được render đầy đủ, giúp trích xuất dữ liệu theo `table_id` và `data-stat`.
- **pandas:** xử lý dữ liệu dạng bảng, chuyển đổi từ `dict` sang `DataFrame`, xử lý missing value và lưu file `.csv`.
- **time:** sử dụng `time.sleep()` để thêm độ trễ giữa các lần tải bảng, tránh bị chặn IP.
- **os:** tạo thư mục đầu ra nếu chưa tồn tại.
- **re:** dùng khi cần xử lý đặc biệt (trích xuất ID từ URL nếu mở rộng thêm).

1.3 Quy trình cào dữ liệu (chi tiết theo mã nguồn)

1.3.1 Bước 1: Đọc cấu hình bảng cần lấy dữ liệu

Từ file `config.py`, chương trình lấy danh sách TABLES, trong đó mỗi bảng được mô tả gồm:

- **url:** đường dẫn cụ thể cho từng loại thống kê (vd: `/en/comps/9/stats/players/` cho thống kê cơ bản).
- **table_id:** ID HTML thực tế của bảng trong DOM (vd: `stats_standard`, `stats_passing...`)
- **fields:** danh sách các chỉ số cần thu thập, mỗi chỉ số là một cặp: (tên cột trong csv, tên `data-stat` trên fbref).

1.3.2 Bước 2: Tải trang bằng Selenium

Với mỗi bảng, chương trình dùng Selenium để:

- Truy cập URL tương ứng (ghép từ `BASE_URL` + `table['url']` + `SEASON_SUFFIX`).
- Dùng `WebDriverWait` để chờ tới khi bảng xuất hiện trong DOM.
- Dùng `driver.page_source` để lấy mã HTML hoàn chỉnh sau khi trang render.

1.3.3 Bước 3: Phân tích HTML bằng BeautifulSoup

- Dùng BeautifulSoup(..., 'html.parser') để phân tích HTML.
- Tìm <table> với id = table_id, sau đó truy cập <tbody>, duyệt từng <tr> đại diện cho từng cầu thủ.
- Với mỗi <tr>, lần lượt tìm các <td> tương ứng với:
 - data-stat="player" → tên cầu thủ
 - data-stat="team" → tên đội bóng
 - và các chỉ số khác từ fields → dữ liệu cụ thể như bàn thắng, kiến tạo, thời gian thi đấu, v.v.
- Dữ liệu mỗi cầu thủ được lưu theo khóa (player_name, team_name) trong dictionary.

1.3.4 Bước 4: Gộp dữ liệu từ nhiều bảng

Sau khi trích xuất xong tất cả các bảng, chương trình gọi combine_data(list_of_dicts) để:

- Gộp toàn bộ dữ liệu theo từng cầu thủ (mỗi cầu thủ có thể xuất hiện nhiều lần nếu chơi cho nhiều đội).
- Nếu một cầu thủ đã có thông tin, các chỉ số mới sẽ update vào từ bảng khác.

1.3.5 Bước 5: Lọc cầu thủ đủ điều kiện (> 90 phút)

- Lặp qua từng dòng dữ liệu đã gộp.
- Lấy cột "Playing Time: minutes", loại bỏ dấu phẩy, chuyển thành số nguyên.
- Chỉ giữ lại cầu thủ có tổng số phút > MIN_MINUTES (được khai báo là 90 trong config.py).

1.3.6 Bước 6: Sắp xếp theo tên

- Dùng hàm get_first_name() để lấy tên đầu của cầu thủ (phần đứng đầu chuỗi họ tên).
- Dữ liệu được sắp xếp lại theo thứ tự bảng chữ cái theo tên đầu.

1.3.7 Bước 7: Chuẩn hóa cột dữ liệu

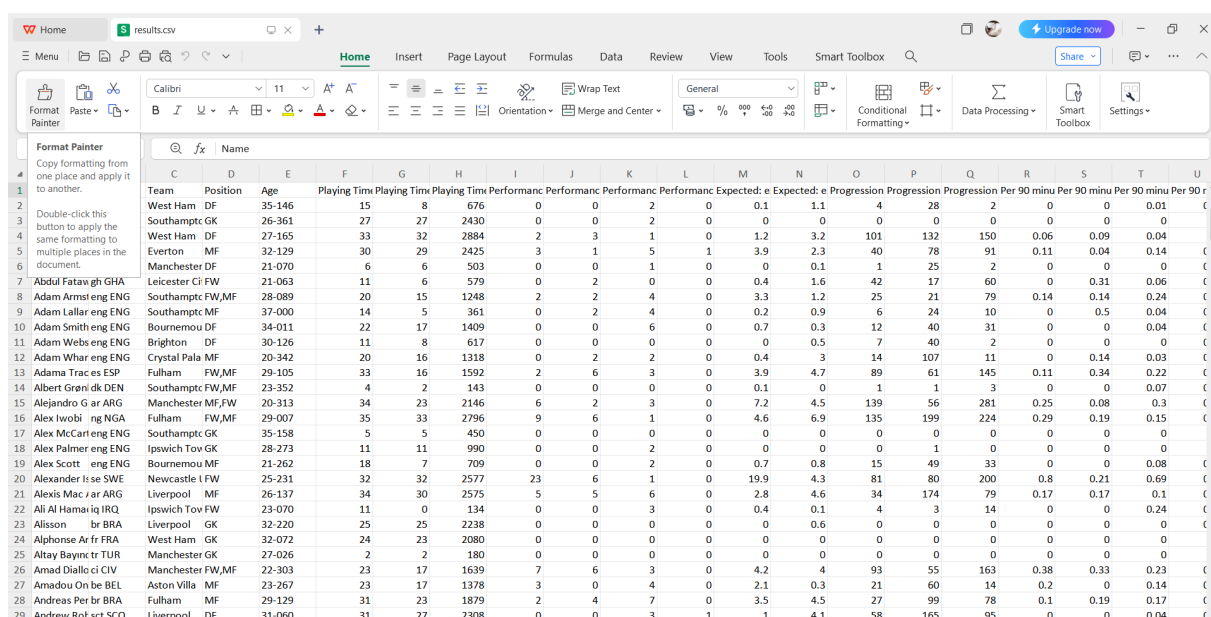
- Duyệt từng hàng trong danh sách lọc được.
- Với mỗi cột trong CSV_COLUMNS, nếu thiếu thì điền giá trị mặc định là "N/a".

1.3.8 Bước 8: Xuất dữ liệu ra tệp CSV

- Dùng `pandas.DataFrame` để chuyển danh sách dict sang bảng.
- Dùng `os.makedirs()` để đảm bảo thư mục `Report/OUTPUT_BAI1` tồn tại.
- Ghi file bằng `df.to_csv(..., encoding="utf-8-sig")` để đảm bảo mở được trong Excel không lỗi font.

1.4 Kết luận

- Tổng số cầu thủ được thu thập và lọc ra trong `results.csv` phụ thuộc vào số phút thi đấu trong toàn mùa, chỉ giữ cầu thủ > 90 phút.
- Các chỉ số thống kê gồm cả cơ bản và nâng cao: Tuổi, Quốc tịch, Thời gian thi đấu, Bàn thắng, Kiến tạo, xG, xAG, các thông số sút, chuyền, thủ môn...
- File kết quả có cấu trúc rõ ràng, phù hợp với yêu cầu đề bài, và là đầu vào cho các bài 2 (thống kê), bài 3 (KMeans clustering), bài 4 (phân tích giá trị chuyển nhượng).
- Chương trình được chia tách hợp lý, dễ đọc, dễ mở rộng. Chỉ cần cập nhật cấu hình trong `config.py` là có thể mở rộng ra các mùa giải hoặc giải đấu khác.



Team	Position	Age	Playing Time	Goals	Assists	xG	xAG	Expected Goals	Expected Assists	Progression	Progression Per 90	Per 90	Per 90	Per 90	Per 90	Per 90	Per 90	Per 90	Per 90
West Ham	DF	35-146	15	8	676	0	0	2	0	0.1	1.1	4	28	2	0	0	0.01	0	0
Southampton	GK	26-361	27	27	2430	0	0	2	0	0	0	0	0	0	0	0	0	0	0
West Ham	DF	27-165	33	32	2884	2	3	1	0	1.2	3.2	101	132	150	0.06	0.09	0.04	0.14	0
Everton	MF	32-129	30	29	2425	3	1	5	1	3.9	2.3	40	78	91	0.11	0.04	0.14	0	0
Manchester	DF	21-070	6	6	503	0	0	1	0	0	0.1	1	25	2	0	0	0	0	0
Abdul Fatawa	gh GHA	Leicester CF	21-063	11	6	579	0	2	0	0.4	1.6	42	17	60	0	0.31	0.06	0	0
Adam Arnsi	eng ENG	Southampton	28-089	20	15	1248	2	2	4	0	3.3	1.2	25	21	79	0.14	0.14	0.24	0
Adam Lallier	eng ENG	Southampton	37-000	14	5	361	0	2	4	0	0.2	0.9	6	24	10	0	0.5	0.04	0
Adam Smith	eng ENG	Bournemouth	34-011	22	17	1409	0	0	6	0	0.7	0.3	12	40	31	0	0	0.04	0
Adam Webb	eng ENG	Brighton	30-126	11	8	617	0	0	0	0	0.5	7	40	2	0	0	0	0	0
Adam Whar	eng ENG	Crystal Palace	20-342	20	16	1318	0	2	2	0	0.4	3	14	107	11	0	0.14	0.03	0
Adama Traoré	ESP	Fulham	29-105	33	16	1592	2	6	3	0	3.9	4.7	89	61	145	0.11	0.34	0.22	0
Albert Grani	dk DEN	Southampton	23-352	4	2	143	0	0	0	0	0.1	0	1	3	0	0	0.07	0	0
Alejandro G	ar ARG	Manchester	20-313	34	23	2146	6	2	3	0	7.2	4.5	139	56	281	0.25	0.08	0.3	0
Alex Iwobi	ng NGA	Fulham	29-007	35	33	2796	9	6	1	0	4.6	6.9	135	199	224	0.29	0.19	0.15	0
Alex McCall	eng ENG	Southampton	35-158	5	5	450	0	0	0	0	0	0	0	0	0	0	0	0	0
Alex Palmer	eng ENG	Ipswich Town	28-273	11	11	990	0	0	2	0	0	0	0	1	0	0	0	0	0
Alex Scott	eng ENG	Bournemouth	21-262	18	7	709	0	0	2	0	0.7	0.8	15	49	33	0	0	0.08	0
Alexander Isak	se SWE	Newcastle	25-231	32	32	2577	23	6	1	0	19.9	4.3	81	80	200	0.8	0.21	0.69	0
Alexis Mac Allister	ar ARG	Liverpool	26-137	34	30	2575	5	5	6	0	2.8	4.6	34	174	79	0.17	0.17	0.1	0
Ali Al Hamayiqi	IRQ	Ipswich Town	23-070	11	0	134	0	0	3	0	0.4	0.1	4	3	14	0	0	0.24	0
Alisson	br BRA	Liverpool	32-220	25	25	2238	0	0	0	0	0	0.6	0	0	0	0	0	0	0
Alphonse Areola	fr FRA	West Ham	32-072	24	23	2080	0	0	0	0	0	0	0	0	0	0	0	0	0
Altay Bayrak	tr TUR	Manchester	27-026	2	2	180	0	0	0	0	0	0	0	0	0	0	0	0	0
Amad Diallo	ci CIV	Manchester	22-303	23	17	1639	7	6	3	0	4.2	4	93	55	163	0.38	0.33	0.23	0
Amadou Onana	be BEL	Aston Villa	23-267	23	17	1378	3	0	4	0	2.1	0.3	21	60	14	0.2	0	0.14	0
Andreas Peris	br BRA	Fulham	29-129	31	23	1879	2	4	7	0	3.5	4.5	27	99	78	0.1	0.19	0.17	0
Andreas Robert	ger GER	Liverpool	31-060	31	27	2308	0	0	3	1	1	4.1	5.8	165	95	0	0	0.04	0

Hình 1.1: Hình ảnh minh họa file results.csv

Chương 2

Phân tích và trực quan hóa dữ liệu

2.1 Phân tích top 3 cầu thủ cao nhất và thấp nhất theo từng chỉ số thống kê

2.1.1 Khái quát về yêu cầu của bài toán

Bài toán yêu cầu xử lý một tệp dữ liệu (results1.csv) chứa thông tin thống kê của các cầu thủ bóng đá. Mục tiêu chính là xác định và liệt kê 3 cầu thủ có điểm số cao nhất (Top 3) và 3 cầu thủ có điểm số thấp nhất (Bottom 3) cho mỗi chỉ số thống kê có trong tệp dữ liệu. Kết quả phân tích này được lưu vào một tệp văn bản có tên là top_3.txt.

2.1.2 Khái quát logic code chính (top_3.py)

Script top_3.py sử dụng thư viện pandas để đọc dữ liệu từ tệp CSV. Sau đó, script lặp qua từng cột thống kê (loại trừ các cột thông tin cơ bản như 'Name', 'Nation', 'Team', 'Position'). Đối với mỗi cột thống kê, code cố gắng chuyển đổi dữ liệu sang dạng số, loại bỏ các giá trị không hợp lệ (NaN). Nếu cột có thể được coi là số và có dữ liệu hợp lệ, code sẽ sắp xếp dữ liệu để tìm ra 3 giá trị cao nhất và 3 giá trị thấp nhất cùng với tên cầu thủ và đội tương ứng. Cuối cùng, kết quả được định dạng và ghi vào tệp top_3.txt.

2.1.3 Xử lý 'N/a' và các giá trị không hợp lệ

Trong quá trình xử lý dữ liệu, các giá trị 'N/a' và các biến thể của chúng được xem là dữ liệu thiếu và được chuyển thành NaN thông qua tham số na_values khi đọc file CSV bằng Pandas. Ngoài ra, các giá trị không thể chuyển đổi sang dạng số cũng được ép về NaN bằng pd.to_numeric(errors='coerce'). Thay vì thay thế các giá trị thiếu bằng một con số cụ thể (như 0), toàn bộ các dòng chứa NaN ở cột cần đánh giá sẽ được loại bỏ trước khi thực hiện việc xếp hạng. Việc này được thực hiện dựa trên các nguyên tắc sau:

- Tính đúng đắn của dữ liệu: NaN thể hiện rằng dữ liệu bị thiếu hoặc không áp dụng, ví dụ như thông số dành riêng cho một vị trí cụ thể không phù hợp với cầu thủ ở vị trí khác. Việc thay thế bằng 0 sẽ làm sai lệch bản chất này.
- Tránh gây hiểu nhầm: Gán giá trị 0 cho dữ liệu thiếu có thể dẫn đến diễn giải sai – ví dụ, hiểu nhầm rằng cầu thủ không ghi bàn, trong khi thực tế dữ liệu chưa được ghi nhận.

- Đảm bảo công bằng khi xếp hạng: Việc chỉ xét các cầu thủ có dữ liệu hợp lệ giúp kết quả Top/Bottom 3 phản ánh đúng hiệu suất thực tế, tránh việc đưa các cá nhân thiếu dữ liệu vào so sánh một cách không chính xác.

Như vậy, việc giữ nguyên NaN và loại bỏ các giá trị thiếu khỏi quá trình xếp hạng là phương pháp hợp lý, đảm bảo tính khách quan và chính xác trong phân tích dữ liệu cầu thủ.

2.1.4 Các bước thực thi

Khởi tạo và chuẩn bị dữ liệu:

- Script nhập các thư viện cần thiết (pandas, os, numpy).
- Đọc dữ liệu từ file results.csv vào DataFrame Pandas, cấu hình để xử lý các biến thể của giá trị thiếu thành NaN.
- Xác định danh sách các cột thống kê (stats_columns) cần phân tích bằng cách loại trừ các cột định danh ('Name', 'Team', 'Position', 'Nation').

Xử lý từng cột và ghi file output:

- Mở file top_3.txt để ghi kết quả.
- Script lặp qua từng cột col trong stats_columns:
 - Ghi Tiêu đề Cột: Ghi tên cột hiện tại vào file.
 - Chuẩn hóa & Lọc Dữ liệu Số: Sử dụng pd.to_numeric(errors='coerce') để ép kiểu cột col sang dạng số, chuyển lỗi thành NaN.
 - Tạo DataFrame tạm df_for_sort và loại bỏ các hàng có giá trị NaN (dropna()) trong cột này. Bước này đảm bảo chỉ dữ liệu số, hợp lệ được sử dụng để xếp hạng.
 - Xếp hạng và Ghi Kết quả:
 - * Nếu df_for_sort chứa dữ liệu số hợp lệ: Dùng sort_values() để sắp xếp DataFrame này theo cột col giảm dần (tìm Top 3) và tăng dần (tìm Bottom 3), sau đó dùng head(3).
 - * Kết quả (Tên, Đội, Chỉ số) và thông tin kiểu dữ liệu được ghi vào file.
 - Ghi Phân cách: Thêm dòng ===... để phân tách kết quả giữa các cột.

Hoàn tất: Sau khi xử lý hết các cột, file top_3.txt được lưu lại, chứa toàn bộ kết quả phân tích.

2.1.5 Kết quả (top_3.txt)

Tệp top_3.txt chứa kết quả phân tích cho từng chỉ số thống kê. Mỗi chỉ số được trình bày rõ ràng với tiêu đề, tiếp theo là danh sách Top 3 và Bottom 3 cầu thủ.

```

--- Age ---
-----

Top 3 Player:
      Name      Team  Age
Łukasz Fabiański West Ham 40.0
      Ashley Young  Everton 39.0
      James Milner  Brighton 39.0

Bottom 3 Player:
      Name      Team  Age
      Mikey Moore  Tottenham 17.0
Chidozie Obi-Martin Manchester Utd 17.0
      Myles Lewis-Skelly  Arsenal 18.0

=====

--- Playing Time: matches played ---
-----

Top 3 Player:
      Name      Team  Playing Time: matches played
Youri Tielemans Aston Villa 35
      Raúl Jiménez  Fulham 35
      Bernd Leno  Fulham 35

Bottom 3 Player:
      Name      Team  Playing Time: matches played
      Ayden Heaven Manchester Utd 2
      Billy Gilmour  Brighton 2
      Neto  Bournemouth 2

=====

```

Hình 2.1: Hình ảnh kết quả file top_3.txt

2.2 Tính toán thống kê mô tả (median, mean, std dev) cho dữ liệu cầu thủ

2.2.1 Giới thiệu

Bài tập này yêu cầu viết một chương trình Python để thực hiện các phép tính thống kê mô tả trên dữ liệu cầu thủ. Cụ thể, chương trình cần tính toán:

- Giá trị trung vị (median) cho mỗi chỉ số thống kê trên toàn bộ tập dữ liệu.
- Giá trị trung bình (mean) và độ lệch chuẩn (standard deviation) cho mỗi chỉ số thống kê, tính toán trên toàn bộ tập dữ liệu và tính riêng cho từng đội (team).

Kết quả được lưu vào tệp **results2.csv** với một định dạng cụ thể, trong đó các hàng đại diện cho toàn bộ ('all') hoặc một đội cụ thể, và các cột đại diện cho các phép tính thống kê (Median, Mean, Std) áp dụng cho từng chỉ số.

2.2.2 Lựa chọn thư viện

- Thư viện **pandas**: Là thư viện chủ đạo, không thể thiếu cho tác vụ này.
- Thư viện **os**: Được sử dụng để quản lý đường dẫn tệp một cách linh hoạt và độc lập với hệ điều hành. `os.path.join` và `os.path.dirname` giúp xác định chính xác vị trí của tệp đầu vào (`results1.csv`) và tệp đầu ra (`results2.csv`) mà không cần mã hóa cứng đường dẫn.

2.2.3 Logic và quy trình thực hiện (`calculating_statistics.py`)

Chương trình `calculating_statistics.py` thực hiện các bước sau:

Bước 1: Chuẩn bị và đọc dữ liệu:

- Sử dụng **os** để xây dựng đường dẫn đến tệp input `results1.csv` và tệp output `results2.csv`.
- Đọc `results.csv` vào pandas DataFrame (`df`).
- Một danh sách `na_values_list` gồm nhiều biến thể của giá trị thiếu (`'N/a'`, `'n/a'`, `''`, ...) được cung cấp cho tham số `na_values` của `pd.read_csv` để đảm bảo nhận diện đúng các giá trị này và chuyển đổi chúng thành `NaN` của pandas, thuận lợi cho các phép tính sau này.

Bước 2: Xác định cột thống kê:

- Định nghĩa danh sách `exclude_cols` chứa các cột không phải là số liệu thống kê cần phân tích (ví dụ: `'Name'`, `'Nation'`, `'Team'`, `'Position'`, `'Age'`).
- Tạo danh sách `stats_columns` bằng cách lọc các cột trong DataFrame, chỉ giữ lại những cột không nằm trong `exclude_cols`.

Bước 3: Tính toán thống kê tổng thể ('all'):

- Chọn các cột trong `stats_columns` từ DataFrame `df`.
- Sử dụng phương thức `.agg(['median', 'mean', 'std'])` để tính đồng thời trung vị, trung bình và độ lệch chuẩn cho từng cột trong `stats_columns`.
- Kết quả trả về có dạng các chỉ số thống kê là hàng, các cột gốc là cột. Sử dụng `.T` (transpose) để chuyển vị, đưa tên các chỉ số gốc (`stats_columns`) thành chỉ số hàng, và `'median'`, `'mean'`, `'std'` thành tên cột, lưu vào `overall_stats`. Điều này giúp truy cập dễ dàng hơn ở bước sau.

Bước 4: Tính toán thống kê theo nhóm ('Team'):

- Sử dụng `df.groupby('Team')` để nhóm DataFrame theo giá trị trong cột `'Team'`.
- Trên đối tượng GroupBy này, chọn các cột `stats_columns`.
- Áp dụng `.agg(['median', 'mean', 'std'])` để tính toán các chỉ số thống kê cho từng đội. Kết quả (`team_stats`) sẽ có cấu trúc đa chỉ số (MultiIndex) ở cả hàng (Team) và cột (Statistic, Metric).

Bước 5: Tái cấu trúc dữ liệu cho đầu ra:

Tạo hàng 'all':

- Khởi tạo một DataFrame rỗng `all_row` với chỉ số là 'all'.
- Lặp qua từng cột thống kê (`col` in `stats_columns`), tạo ra các cột mới trong `all_row` với tên theo định dạng `f'Median of {col}'`, `f'Mean of {col}'`, `f'Std of {col}'` và gán giá trị tương ứng lấy từ `overall_stats` đã tính ở Bước 3.

Tạo các hàng 'Team':

- Khởi tạo một DataFrame `team_results` với chỉ số là tên các đội (lấy từ `team_stats.index`).
- Lặp qua từng cột thống kê (`col` in `stats_columns`), tạo ra các cột mới trong `team_results` với tên định dạng tương tự như trên. Giá trị được lấy từ `team_stats` bằng cách truy cập qua chỉ số đa cấp, ví dụ: `team_stats[(col, 'median')]` để lấy cột trung vị của chỉ số `col`.

Kết hợp Kết quả:

- Sử dụng `pd.concat([all_row, team_results])` để nối DataFrame chứa hàng 'all' và DataFrame chứa các hàng của từng đội lại với nhau theo chiều dọc, tạo thành DataFrame cuối cùng `final_results` có cấu trúc đúng yêu cầu.

Bước 6: Lưu Kết quả:

- Xuất DataFrame `final_results` ra tệp `results2.csv` bằng phương thức `.to_csv()`.
- Tham số `index=True` được sử dụng (mặc định) để lưu chỉ số của DataFrame (chính là 'all' hoặc tên đội) vào cột đầu tiên của tệp CSV.

2.2.4 Kết quả

Chương trình đã chạy thành công và tạo ra tệp `results2.csv`. Tệp `results2.csv` chứa các kết quả thống kê tổng hợp theo định dạng yêu cầu:

- Hàng: Hàng đầu tiên có chỉ số là 'all', đại diện cho thống kê trên toàn bộ cầu thủ. Các hàng tiếp theo có chỉ số là tên của từng đội, đại diện cho thống kê tính riêng cho cầu thủ của đội đó.
- Cột: Các cột được đặt tên theo mẫu "Metric of Statistic", ví dụ: "Median of Performance: goals", "Mean of Performance: goals", "Std of Performance: goals", "Median of Performance: assists", v.v., bao gồm tất cả các chỉ số trong `stats_columns`.
- Giá trị: Các ô chứa giá trị trung vị, trung bình hoặc độ lệch chuẩn tương ứng đã được tính toán. Các giá trị NaN (nếu có, ví dụ: độ lệch chuẩn của nhóm chỉ có 1 cầu thủ) sẽ được biểu diễn dưới dạng ô trống trong tệp CSV.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
10	Everton	24	22.4090909	9.28726260	15	17.4545454	10.6490655	1357.5	1565.22727	904.269183	1	1.5	2.13251472	1	1.11269728	3	3.40909090	2.17472693	0	0.090909	
11	Fulham	27	24.5	9.32865529	17.5	17.4545454	11.4962821	1620.5	1568.27272	963.243685	0.5	2.22727272	3.26499206	1	1.90909090	2.56179049	2.5	3.40909090	2.98662240	0	0.090909
12	Ipswich Tow	18.5	18.2	9.01493014	11.5	12.8333333	9.68498066	984.5	1146.76666	798.991290	0	1.13333333	2.31536596	0	0.83333333	1.08543121	2	2.96666666	2.64553408	0	0.166666
13	Leicester Ci	21.5	20.2307692	9.71723290	15	14.8076923	9.96802580	1408	1329.19230	824.506301	0	1.11538461	2.00652780	0	0.84615384	1.18969938	2	3.11538461	2.64313333	0	
14	Liverpool	28	25.1904761	8.92535180	20	18.3333333	11.9219685	1717	1643.09523	977.499048	1	3.80952380	6.49321990	2	2.85714285	3.99106144	3	3.04761904	2.26883649	0	0.095228
15	Manchester	23	19.88	8.97366517	17	15.36	8.62592990	1494	1381.36	766.282306	1	2.64	4.41474801	0	1.92	2.56450125	2	2.32	1.74928556	0	
16	Manchester	20	17.6666666	11.4540561	11.5	12.8333333	10.7801968	1071	1149.66666	931.015327	0	1.33333333	2.26416359	0	0.93333333	1.99885024	2	2.6	2.47191116	0	
17	Newcastle U	27	23.2179113	10.1442169	14	16.7391304	12.5886579	1503	1502.86956	1050.91130	0	2.78260869	5.18701632	1	2.08695652	2.84306242	2	2.60869565	3.02623574	0	0.043478
18	Nott'ham Fi	30	24.5	10.8879575	19	17.5	13.3014499	1804	1572.68181	1104.92513	1	2.40909090	4.15943094	1	1.77272727	2.48676148	3	3.72727272	3.23936822	0	0.090909
19	Southampton	20	18.6551724	10.3794985	13	13.2413793	9.84035122	1178	1184.31034	856.237287	0	0.82758620	1.07134646	0	0.51724137	0.82897051	2	2.86206896	2.94865753	0	0.103448
20	Tottenham	22	19.3333333	9.23205119	16	14.2222222	8.09241493	1300	1277.77777	694.565073	1	2.22222222	3.25024653	1	1.70370370	2.38286858	2	2.44444444	2.32599578	0	0.037037
21	West Ham	20	21.56	8.52975185	14	15.4	10.8627804	1070	1381.52	914.388507	0	1.48	2.6	1	0.96	1.51327459	2	2.96	2.65329983	0	
22	Wolves	26	22.7826086	8.92369940	15	16.6521739	10.9070815	1364	1495.21739	875.622477	1	2.17391304	3.93876046	1	1.69565217	2.22454862	2	3.21739130	2.74617952	0	0.086957

Hình 2.2: Hình ảnh 1 phần của file results2.csv

2.3 Trực quan hóa phân phối thống kê cầu thủ bằng biểu đồ histogram

2.3.1 Mục tiêu

Mục tiêu chính của script `plot_stat_histograms.py` là trực quan hóa và phân tích sự phân phối của một tập hợp các chỉ số thống kê quan trọng của cầu thủ bóng đá. Cụ thể, script nhằm:

- Tạo biểu đồ histogram cho từng chỉ số thống kê được chọn (`stats_to_plot`) để thể hiện sự phân phối của chỉ số đó trên toàn bộ giải đấu (tất cả cầu thủ).
- Tạo biểu đồ histogram riêng cho từng đội bóng, thể hiện sự phân phối của từng chỉ số thống kê trong nội bộ đội đó.
- Lưu các biểu đồ này dưới dạng file ảnh (PNG) để dễ dàng xem xét, chia sẻ và phân tích sau này.
- Cung cấp cái nhìn trực quan về hình dạng phân phối (ví dụ: đối xứng, lệch trái, lệch phải), xu hướng trung tâm, và độ phân tán của các chỉ số thống kê, cả ở cấp độ giải đấu và cấp độ đội bóng.

Các chỉ số thống kê cụ thể được phân tích trong script này bao gồm: 'Performance: goals', 'Performance: assists', 'Standard: shoots on target percentage (SoT%)', 'Blocks: Int', 'Performance: Recov', 'Challenges: Att'.

2.3.2 Lựa chọn Thư viện

Script sử dụng một số thư viện Python phổ biến cho xử lý dữ liệu và trực quan hóa:

- **pandas**: Thư viện cốt lõi để đọc dữ liệu từ file CSV (`results1.csv`), lưu trữ dữ liệu dưới dạng DataFrame, và thực hiện các thao tác xử lý, lọc, và lựa chọn dữ liệu cần thiết cho việc vẽ biểu đồ. Nó cũng xử lý các giá trị thiếu (NaN).
- **matplotlib.pyplot**: Thư viện nền tảng cho việc tạo biểu đồ trong Python. Nó được sử dụng để tạo khung hình (figure), các trục (axes), đặt tiêu đề, nhãn, và quản lý layout của biểu đồ, đặc biệt khi vẽ nhiều biểu đồ con (subplots) cho các đội.

- **seaborn**: Thư viện xây dựng trên matplotlib, cung cấp giao diện cấp cao hơn để vẽ các biểu đồ thống kê hấp dẫn và giàu thông tin. Cụ thể, `seaborn.histplot` được dùng để vẽ biểu đồ histogram, bao gồm cả đường ước lượng mật độ xác suất giúp làm mịn và hình dung rõ hơn hình dạng phân phối (KDE - Kernel Density Estimate). `seaborn.set_theme` dùng để thiết lập phong cách thẩm mỹ chung cho các biểu đồ.
- **numpy**: Thư viện tính toán khoa học, được sử dụng trong hàm `calculate_fd_bins` để thực hiện các phép toán cần thiết (như căn bậc ba `np.cbrt`, logarit cơ số 2 `np.log2`) cho việc tính toán số lượng bins tối ưu theo quy tắc Freedman-Diaconis và Sturges.
- **os**: Thư viện cung cấp các hàm tương tác với hệ điều hành, chủ yếu được dùng để quản lý đường dẫn file (lấy đường dẫn thư mục hiện tại, nối đường dẫn), tạo thư mục lưu trữ (`stat_histograms`), và đảm bảo đường dẫn hợp lệ.
- **math**: Thư viện toán học cơ bản, được sử dụng cho các hàm như `math.ceil` để làm tròn lên khi tính toán số lượng hàng/cột cho subplot và số lượng bins.

2.3.3 Các bước thực hiện

Định nghĩa hàm hỗ trợ:

- **ensure_dir**: Đảm bảo thư mục đầu ra tồn tại.
- **sanitize_filename**: Làm sạch tên chỉ số thống kê để tạo tên file hợp lệ.
- **calculate_fd_bins**: Tính số lượng bins tối ưu cho histogram dựa trên quy tắc Freedman-Diaconis (ưu tiên) hoặc Sturges (dự phòng), giúp biểu đồ phản ánh tốt hơn cấu trúc dữ liệu thực tế và tránh việc chọn số bins tùy tiện. Có giới hạn số bins tối đa để tránh quá chi tiết.

Định nghĩa hàm vẽ biểu đồ:

- **plot_overall_hist**:
 - Nhận DataFrame, tên chỉ số thống kê, và thư mục đầu ra.
 - Lọc dữ liệu cho chỉ số thống kê, chuyển đổi sang dạng số và loại bỏ giá trị thiếu (NaN).
 - Tính số bins tối ưu bằng `calculate_fd_bins`.
 - Sử dụng `seaborn.histplot` để vẽ histogram phân phối tổng thể, bật KDE.
 - Đặt tiêu đề, nhãn trục và lưu biểu đồ vào file PNG với tên được chuẩn hóa.
- **plot_team_hist**:
 - Nhận DataFrame, tên chỉ số, thư mục đầu ra, tên cột đội, và số đội trên mỗi hình.
 - Lấy danh sách các đội duy nhất và chia thành các nhóm nhỏ (theo `TEAMS_PER_PLOT`).
 - Lặp qua từng nhóm đội:
 - * Tạo một figure mới với các subplots (lưới biểu đồ con).

- * Lọc dữ liệu chỉ chứa các đội trong nhóm hiện tại.
- * Tính số bins tối ưu dựa trên phạm vi dữ liệu của các đội trong nhóm này. Điều này giúp các histogram trong cùng một figure có cùng cách chia khoảng giá trị.
- * Xác định giới hạn trục x (`x_min`, `x_max`) chung cho tất cả các subplot trong figure hiện tại để dễ so sánh.
- * Vẽ histogram cho từng đội trên một subplot riêng biệt bằng `seaborn.histplot`, sử dụng số bins và phạm vi bin (`binrange`) đã tính.
- * Đặt tiêu đề con là tên đội, ẩn nhãn trục để tránh rối mắt, giới hạn trục x thống nhất.
- * Ẩn các subplot thừa.
- * Đặt tiêu đề chính cho figure (bao gồm tên chỉ số và nhóm đội).
- * Lưu figure chứa histogram của nhóm đội vào file PNG.

Hàm thực thi chính (if `__name__` == '`__main__`':)

- Thiết lập theme cho seaborn.
- Định nghĩa danh sách các chỉ số cần vẽ (`stats_to_plot`).
- Xác định đường dẫn và đọc file `results.csv` vào DataFrame.
- Đảm bảo thư mục đầu ra tồn tại.
- Lặp qua từng chỉ số trong `stats_to_plot`:
 - Gọi `plot_overall_hist` để vẽ biểu đồ tổng thể.
 - Gọi `plot_team_hist` để vẽ biểu đồ cho từng đội (được nhóm lại).

2.3.4 Hạn chế của việc sử dụng biểu đồ Histogram

Mặc dù biểu đồ histogram là một công cụ trực quan hóa mạnh mẽ trong phân tích dữ liệu thống kê cầu thủ, cho phép chúng ta nhanh chóng nắm bắt phân phối của các chỉ số (ví dụ: số bàn thắng, thời gian thi đấu, số kiến tạo), xác định giá trị ngoại lai, và so sánh các cầu thủ hoặc đội, cần lưu ý rằng phương pháp này cũng có những hạn chế nhất định. Để tránh những kết luận sai lệch hoặc phiến diện, chúng ta cần xem xét các yếu tố sau:

- **Ảnh hưởng của chất lượng dữ liệu:** Độ tin cậy của biểu đồ histogram phụ thuộc trực tiếp vào tính đầy đủ và chính xác của dữ liệu đầu vào (trong tệp `results.csv`). Việc loại bỏ các giá trị thiếu (NaN) trong quá trình xử lý là cần thiết để đảm bảo tính chính xác của biểu đồ. Tuy nhiên, nếu có quá nhiều dữ liệu bị thiếu hoặc việc thiếu dữ liệu không xảy ra ngẫu nhiên, hình dạng của phân phối có thể bị bóp méo, dẫn đến những đánh giá chủ quan và thiếu chính xác về xu hướng chung.
- **Tính nhạy cảm với số lượng khoảng (bins):** Mặc dù quy tắc Freedman-Diaconis được áp dụng để xác định số lượng khoảng tối ưu, nhằm thể hiện chi tiết dữ liệu một cách tốt nhất, hình dạng của histogram vẫn có thể thay đổi đáng kể khi số lượng khoảng thay đổi. Thêm vào đó, việc áp đặt một giới hạn tối đa cho số lượng khoảng (thông qua biến `MAX_FD_BINS`) mang tính chủ quan và có khả năng làm giảm độ phân giải của biểu đồ, đặc biệt khi phân phối dữ liệu có độ phân tán lớn.

2.4 Phân tích hiệu suất các đội bóng Premier League 2024–2025

2.4.1 Giới thiệu

Để có được cái nhìn sâu sắc về phong độ của các đội bóng trong mùa giải Premier League 2024–2025, phân tích này được xây dựng xung quanh hai mục tiêu chính. Thứ nhất, là xác định đội bóng nào đang dẫn đầu ở từng hạng mục thống kê. Thứ hai, là đưa ra đánh giá về đội bóng nào có tổng thể màn trình diễn ấn tượng nhất. Để thực hiện điều này, chúng ta dựa vào thông tin được xử lý bởi hai thành phần phần mềm quan trọng:

- **highest_stats_team.py**: Đây là trái tim của quá trình phân tích - một script Python được thiết kế để tự động hóa mọi thứ. Nó đảm nhận các nhiệm vụ từ việc đọc dữ liệu đầu vào, sắp xếp các chỉ số thành các loại khác nhau (ví dụ: chỉ số tích cực, tiêu cực, hoặc các chỉ số không liên quan), và sau đó xác định đội bóng nào có thành tích tốt nhất (hoặc kém nhất, tùy trường hợp) ở mỗi hạng mục. Cuối cùng, nó tổng hợp tất cả các thông tin này để đưa ra kết luận về đội bóng xuất sắc nhất của mùa giải.
- **config.py**: File này đóng vai trò như một bảng điều khiển, cho phép người dùng tùy chỉnh các yếu tố khác nhau của quá trình phân tích. Chẳng hạn, người dùng có thể thay đổi đường dẫn đến dữ liệu, các quy tắc để xử lý dữ liệu bị thiếu, hoặc cách thức phân loại các chỉ số. Sự linh hoạt này giúp cho phần mềm có thể được sử dụng lại một cách dễ dàng, giảm thiểu công sức bảo trì, và cho phép phân tích dữ liệu từ các mùa giải khác nhau hoặc các nguồn khác nhau.

Cách tiếp cận này không chỉ giúp tiết kiệm thời gian và công sức mà còn mang lại sự linh hoạt và khả năng mở rộng cho các phân tích trong tương lai.

2.4.2 Ý tưởng chính

- **Xác định các đội dẫn đầu theo từng chỉ số**: Chương trình được thiết kế để phân chia các chỉ số thống kê thành ba nhóm chính (chỉ số có lợi, chỉ số bất lợi, và chỉ số bị bỏ qua) dựa trên các quy tắc được quy định trong file **config.py**. Đối với mỗi chỉ số, dữ liệu được chuyển đổi sang dạng số, và sau đó chương trình sẽ tìm kiếm giá trị cao nhất (đối với chỉ số có lợi) hoặc giá trị thấp nhất (đối với chỉ số bất lợi). Đội bóng nào đạt được giá trị này sẽ được xác định là đội dẫn đầu ở hạng mục đó. Để minh họa, chương trình có thể xác định đội nào ghi được nhiều bàn thắng nhất, đội nào có tỷ lệ chuyền bóng thành công cao nhất, hoặc đội nào nhận ít thẻ đỏ nhất.
- **Xếp hạng các đội dựa trên hiệu suất tổng thể**: Chương trình sẽ đếm số lượng các chỉ số mà mỗi đội đứng đầu, chia riêng cho các chỉ số có lợi và các chỉ số bất lợi. Sau đó, nó sẽ tính toán một "điểm số hiệu suất" cho từng đội bằng cách cộng số lần dẫn đầu ở cả hai loại chỉ số. Đội nào có điểm số hiệu suất cao nhất sẽ được coi là đội có thành tích tốt nhất trong mùa giải.

2.4.3 Quy trình thực thi

Quá trình phân tích được thực hiện theo một chuỗi các bước được kiểm soát bởi script **highest_stats_team.py**:

1. Thiết lập và nạp các thông số cấu hình:

Các thông số cấu hình được đọc từ file `config.py` thông qua một hàm có tên `load_config`. Hàm này có nhiệm vụ đọc các biến cấu hình khác nhau và tạo ra các tập hợp (set) từ các biến `IGNORE_STATS`, `BAD_STATS`, và `GOOD_STATS`. Các tập hợp này được sử dụng để phân loại các chỉ số thống kê trong các bước tiếp theo.

2. Xây dựng đường dẫn và đọc dữ liệu từ file CSV:

Chương trình tạo ra đường dẫn đầy đủ đến file `results2.csv`. Dữ liệu từ file CSV được đọc vào một DataFrame (một loại bảng dữ liệu) bằng cách sử dụng thư viện `pandas`. Các quy tắc xử lý các giá trị bị thiếu được lấy từ biến `NA_VALUES` trong file `config.py`.

3. Xử lý dữ liệu ban đầu trong DataFrame:

Cột chứa thông tin về tên của các đội bóng được xác định (cột đầu tiên được coi là mặc định). Tên của tất cả các đội được chuyển đổi thành chữ thường, và các dòng dữ liệu có tên đội là "all" sẽ bị loại bỏ khỏi DataFrame.

4. Tìm kiếm các đội dẫn đầu cho từng chỉ số thống kê hợp lệ (Hàm main):

Bước 1: Chọn các chỉ số thống kê để đưa vào phân tích:

Chương trình tạo ra một danh sách bao gồm các cột có tiền tố "Mean of "(ngoại trừ cột chứa tên đội). Đối với mỗi cột trong danh sách này: Tên thực tế của chỉ số (bỏ phần "Mean of ") được tách ra. Chỉ số được gán một "loại" (sử dụng hàm `classify_stat`): 'ignore' (bỏ qua), 'bad' (chỉ số không mong muốn), 'good' (chỉ số mong muốn), hoặc 'uncategorized' (chưa được phân loại). Việc gán loại này dựa trên các tập hợp từ khóa đã được tải ở bước trước. Nếu chỉ số không bị bỏ qua, dữ liệu trong cột đó được chuyển đổi sang dạng số. Chỉ các cột có ít nhất một giá trị số hợp lệ mới được giữ lại để phân tích sâu hơn.

Bước 2: Phân tích chi tiết từng chỉ số đã chọn:

Đối với mỗi chỉ số được giữ lại ở bước trước: Dữ liệu trong cột chỉ số được chuyển đổi sang dạng số, và bất kỳ giá trị không hợp lệ nào (NaN) đều bị loại bỏ. Chương trình xác định giá trị "tốt nhất": Nếu chỉ số thuộc loại 'good' hoặc 'uncategorized': chương trình tìm kiếm giá trị lớn nhất. Nếu chỉ số thuộc loại 'bad': chương trình tìm kiếm giá trị nhỏ nhất. Nếu chương trình không tìm thấy giá trị tốt nhất (ví dụ: cột chỉ chứa các giá trị NaN), nó sẽ ghi lại một thông báo lỗi. Ngược lại (nếu giá trị tốt nhất được tìm thấy): Chương trình xác định (các) đội bóng nào đạt được giá trị tốt nhất này. Thông tin về (các) đội dẫn đầu, giá trị tốt nhất (được giữ nguyên định dạng ban đầu để hiển thị), và loại của chỉ số được lưu trữ trong một cấu trúc dữ liệu gọi là dictionary.

Bước 3: Trả về kết quả tổng hợp:

Hàm trả về một dictionary chứa tất cả các kết quả phân tích cho từng chỉ số.

5. Trình bày kết quả phân tích chi tiết:

Các kết quả được sắp xếp theo tên của chỉ số thống kê. Đối với mỗi chỉ số đã được phân tích, chương trình định dạng giá trị (sử dụng hàm `format_value` để xử lý số nguyên, số thực, và chuỗi "N/A") và in ra các thông tin sau: tên của chỉ số, loại của chỉ số (GOOD, BAD), đội dẫn đầu, và giá trị tốt nhất mà đội đó đạt được.

6. Tính toán và hiển thị bảng xếp hạng tổng thể:

Bước 1: Đếm số lượng lần dẫn đầu:

Đối với mỗi chỉ số có kết quả hợp lệ trong cấu trúc dữ liệu `analysis_results`: Nếu đội dẫn đầu chỉ số thuộc loại 'good', số lượng lần dẫn đầu cho đội đó trong cấu trúc dữ liệu `good_lead_counts` sẽ được tăng lên. Nếu đội dẫn đầu chỉ số thuộc loại 'bad', số lượng lần dẫn đầu cho đội đó trong cấu trúc dữ liệu `bad_lead_counts` sẽ được tăng lên.

Bước 2: Tính toán điểm số hiệu suất tổng thể:

Chương trình thu thập danh sách tất cả các đội bóng khác nhau có trong DataFrame. Điểm số ban đầu của mỗi đội được đặt là 0. Điểm số hiệu suất tổng thể cho mỗi đội được

tính bằng cách cộng tổng số lần dẫn đầu ở các chỉ số 'good' và 'bad'.

Bước 3: Sắp xếp và hiển thị bảng xếp hạng:

Các đội bóng được sắp xếp theo thứ tự giảm dần của điểm số hiệu suất tổng thể. Nếu có hai đội có cùng điểm số, chúng sẽ được sắp xếp theo thứ tự bảng chữ cái của tên đội. Bảng xếp hạng cuối cùng được in ra, bao gồm thứ hạng của mỗi đội, tên đội, và tổng số lần dẫn đầu.

2.4.4 Kết quả

Khi chương trình `highest_stats_team.py` hoàn thành việc chạy, nó sẽ tạo ra hai phần thông tin chính, được hiển thị trực tiếp trên màn hình. Những kết quả này cung cấp cái nhìn toàn diện về hiệu suất của từng đội ở mỗi chỉ số thống kê được phân tích, cũng như một bảng xếp hạng tổng hợp các đội dựa trên số lần họ đạt được vị trí dẫn đầu ở các chỉ số đó.

Ví dụ về kết quả phân tích chi tiết cho từng chỉ số:

```
Teams with the highest (or lowest for bad stats) value for each statistic:  
Performance: goals [GOOD]: Liverpool (value: 28)  
Performance: assists [GOOD]: Liverpool (value: 18)  
Performance: yellow cards [BAD]: Leicester City (value: 0)  
Performance: red cards [BAD]: West Ham (value: 0)  
Expected: expected goals (xG) [GOOD]: Liverpool (value: 24.0)  
Expected: expected Assist Goals (xAG) [GOOD]: Liverpool (value: 13.2)  
Progression: PrgC [GOOD]: Manchester City (value: 189)  
Progression: PrgP [GOOD]: Manchester Utd (value: 281)  
Progression: PrgR [GOOD]: Liverpool (value: 440)  
Per 90 minutes: GlS [GOOD]: Aston Villa (value: 0.99)
```

Hình 2.3: Hình ảnh ví dụ kết quả phân tích chi tiết cho từng chỉ số

Ví dụ về bảng xếp hạng tổng hợp các đội:

```
Summary ranking (number of times leading in statistics):  
Liverpool: 17 top statistics  
West Ham: 5 top statistics  
Southampton: 5 top statistics  
Manchester City: 4 top statistics  
Manchester Utd: 4 top statistics  
Chelsea: 4 top statistics  
Brighton: 4 top statistics  
Brentford: 4 top statistics  
Bournemouth: 2 top statistics
```

Hình 2.4: Hình ảnh ví dụ về bảng xếp hạng tổng hợp các đội

Dữ liệu thống kê cho thấy Liverpool nổi lên như một đội bóng có sức mạnh và sự toàn diện đáng kinh ngạc. Họ không chỉ xuất sắc ở một vài khía cạnh nhất định mà còn duy

trì được phong độ ổn định ở hầu hết các thông số quan trọng của bóng đá hiện đại. Đội bóng này sở hữu khả năng tấn công mạnh mẽ, sự vượt trội trong việc kiểm soát bóng, cùng với một hệ thống phòng ngự vững chắc – tạo nên một tập thể thi đấu ăn ý và hiệu quả. Trên thực tế, Liverpool đã chắc chắn giành chức vô địch Premier League mùa giải 2024–2025 trước khi mùa giải kết thúc 4 vòng đấu, kết thúc với 82 điểm và bỏ xa đội á quân Arsenal với một khoảng cách an toàn. Đây là một minh chứng rõ ràng cho thấy, khi một đội bóng vận hành tốt cả về mặt chuyên môn và chiến lược, thành công của họ sẽ được phản ánh một cách rõ rệt qua các số liệu thống kê và kết quả trên sân cỏ.

Chương 3

Phân cụm Cầu thủ bằng K-Means và PCA

3.1 Giới thiệu và chuẩn bị dữ liệu

3.1.1 Giới thiệu

Báo cáo này trình bày chi tiết về việc ứng dụng thuật toán phân cụm K-means để phân loại các cầu thủ bóng đá thành các nhóm dựa trên số liệu thống kê về hiệu suất thi đấu của họ. Mục tiêu chính của quá trình này là:

- Xác định số lượng cụm tối ưu (tức là số lượng nhóm tốt nhất) mà các cầu thủ nên được chia vào.
- Tiến hành phân cụm cầu thủ dựa trên số lượng cụm đã xác định.
- Trực quan hóa các cụm này trên biểu đồ hai chiều, sử dụng kỹ thuật giảm chiều dữ liệu Principal Component Analysis (PCA).

3.1.2 Chuẩn bị dữ liệu

Để chuẩn bị cho việc phân tích, dữ liệu về các chỉ số của cầu thủ đã được xử lý qua các bước sau:

- **Tải dữ liệu:** Dữ liệu được tải từ một tập tin CSV (Comma Separated Values), đây là định dạng phổ biến để lưu trữ dữ liệu bảng.
- **Chọn các cột dữ liệu liên quan:** Các cột chứa thông tin thống kê về hiệu suất của cầu thủ (ví dụ: số bàn thắng, số đường kiến tạo, số lần tắc bóng, v.v.) được chọn để đưa vào phân tích. Các cột không liên quan đến thống kê (ví dụ: tên cầu thủ, quốc tịch, tên đội) và cột thông tin về đội bóng bị loại bỏ.
- **Xử lý giá trị thiếu:** Trong quá trình thu thập dữ liệu, có thể có những trường hợp một số thông tin bị thiếu. Để đảm bảo tính toàn vẹn của phân tích, các giá trị thiếu này được điền vào bằng giá trị trung bình của cột chứa chúng.
- **Chuẩn hóa dữ liệu:** Các chỉ số thống kê khác nhau có thể có thang đo và đơn vị khác nhau. Để đảm bảo rằng mọi chỉ số đều có đóng góp ngang nhau vào quá trình phân cụm, dữ liệu được "chuẩn hóa" bằng StandardScaler. Quá trình này chuyển đổi dữ liệu sao cho nó có giá trị trung bình bằng 0 và độ lệch chuẩn bằng 1.

3.2 Phân cụm và giảm chiều dữ liệu

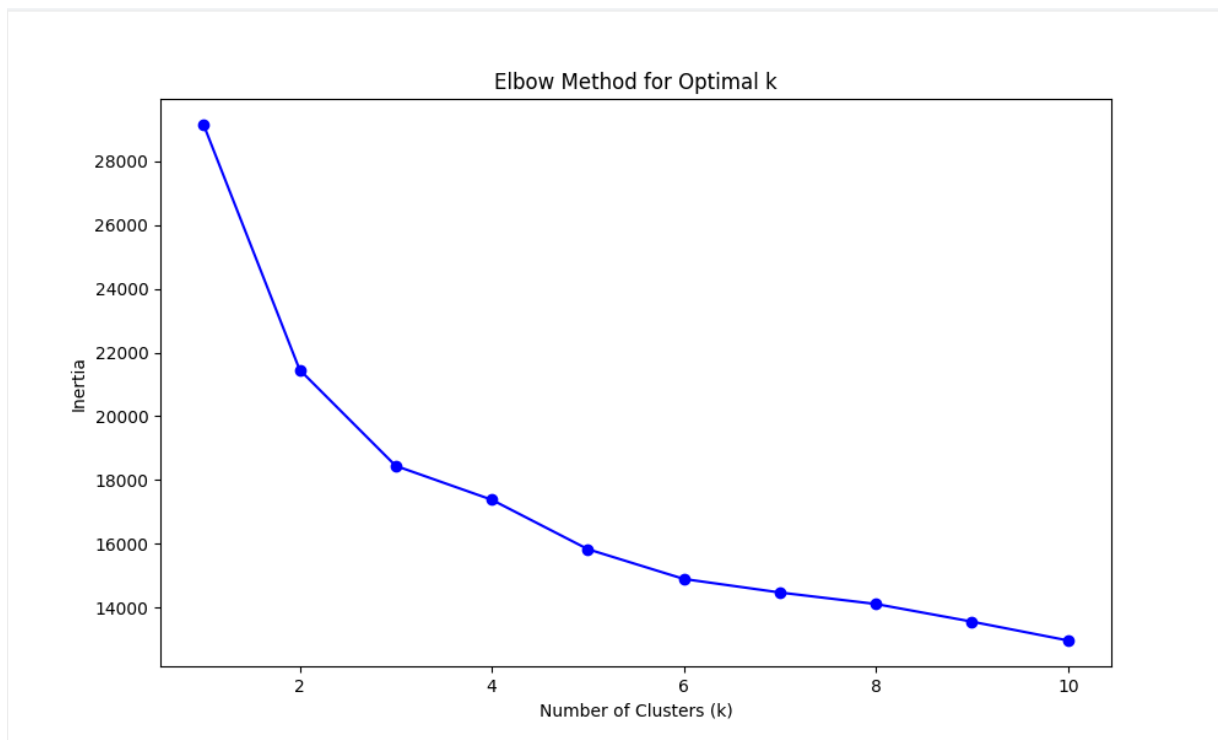
3.2.1 Xác định số lượng cụm k tối ưu

Để xác định số lượng cụm (k) tối ưu, phương pháp Elbow (khủy tay) đã được sử dụng. Phương pháp này hoạt động như sau:

- **Tính quán tính (Inertia):** Với mỗi giá trị k (số lượng cụm), thuật toán K-means được chạy để phân chia dữ liệu thành các cụm. "Quán tính" là tổng của bình phương khoảng cách từ mỗi điểm dữ liệu đến tâm cụm gần nhất của nó. Quán tính càng nhỏ thì các điểm trong cụm càng gần nhau, tức là cụm càng chặt chẽ.
- **Vẽ biểu đồ Elbow:** Một biểu đồ được vẽ, với trục x là số lượng cụm (k) và trục y là giá trị quán tính tương ứng.
- **Tìm điểm "khủy tay":** Hình dạng của biểu đồ thường giống như một cánh tay, và điểm "khủy tay" là điểm mà tại đó độ dốc của đường cong thay đổi đột ngột. Điểm này cho thấy rằng việc tăng số lượng cụm sau điểm đó không làm giảm quán tính đi nhiều, tức là không cải thiện đáng kể độ chặt chẽ của các cụm.

Trong nghiên cứu này:

- Các giá trị k được thử nghiệm nằm trong khoảng từ 1 đến 10. Giá trị tối đa là 10 được chọn để giới hạn độ phức tạp tính toán, đồng thời vẫn cho phép khám phá một số lượng cụm hợp lý.
- Biểu đồ Elbow (`elbow_plot.png`) cho thấy rõ sự thay đổi độ dốc lớn nhất tại $k = 6$.



Hình 3.1: Hình ảnh biểu đồ Elbow

- Do đó, số lượng cụm tối ưu được chọn là 6.

Phân cụm K-Means

- Thuật toán K-means được chạy với số lượng cụm (k) được đặt thành 6.
- Thuật toán này chia các cầu thủ thành 6 cụm khác nhau, sao cho mỗi cầu thủ được gán cho cụm có tâm gần nhất với mình.
- Các cụm này được gán nhãn từ 0 đến 5.

3.2.2 Giảm chiều dữ liệu với PCA

- Dữ liệu về cầu thủ ban đầu có nhiều chiều (mỗi chiều tương ứng với một chỉ số thống kê). Để có thể biểu diễn trực quan trên biểu đồ 2D, cần giảm số chiều của dữ liệu.
- Kỹ thuật Principal Component Analysis (PCA) được sử dụng để giảm số chiều xuống còn 2. PCA tìm ra hai "thành phần chính"(PC1 và PC2) là hai hướng kết hợp các chỉ số ban đầu sao cho giữ lại được nhiều thông tin nhất của dữ liệu.
- "Tỷ lệ phương sai được giải thích"(explained variance ratio) cho biết phần trăm thông tin (phương sai) của dữ liệu gốc được giữ lại trong hai thành phần chính này. Giá trị này càng cao thì biểu đồ 2D càng thể hiện chính xác dữ liệu gốc.

3.2.3 Trực quan hóa các cụm

- Một biểu đồ phân tán 2D (`player_clusters.png`) được tạo, với PC1 là trục x và PC2 là trục y.
- Mỗi điểm trên biểu đồ đại diện cho một cầu thủ.
- Màu sắc của mỗi điểm cho biết cụm mà cầu thủ đó được gán.
- Màu sắc tương ứng với các nhãn cụm như sau:
 - Xanh lam: Cụm 0
 - Cam: Cụm 1
 - Xanh lá cây: Cụm 2
 - Đỏ: Cụm 3
 - Tím: Cụm 4
 - Nâu: Cụm 5

3.2.4 Phân tích cụm

Để hiểu rõ hơn về ý nghĩa của từng cụm, một phân tích chi tiết đã được thực hiện:

- **Cụm 0 (Xanh lam): Các hậu vệ biên chuyên bóng tốt**
 - *Vị trí:* Cụm này chủ yếu bao gồm các hậu vệ cánh và hậu vệ biên.
 - *Đặc điểm nổi bật:*

- * Các cầu thủ trong cụm này có xu hướng thực hiện số lượng lớn các đường chuyền, đặc biệt là các đường chuyền ngắn và trung bình, với độ chính xác cao.
- * Họ cũng thường xuyên tham gia vào việc luân chuyển bóng và đưa bóng lên phía trên sân.
- * Các chỉ số phòng ngự của họ ở mức trung bình, cho thấy họ có đóng góp vào cả tấn công và phòng ngự.

• **Cụm 1 (Cam): Các tiền vệ trung tâm toàn diện**

- *Vị trí:* Cụm này tập trung vào các tiền vệ trung tâm.
- *Đặc điểm nổi bật:*
 - * Các cầu thủ ở cụm này có sự cân bằng giữa khả năng tấn công và phòng ngự.
 - * Họ tham gia vào cả việc thu hồi bóng, đánh chặn, và phát động tấn công, chuyền bóng ở khu vực giữa sân.
 - * Họ có khả năng chuyền bóng tốt, cả ở cự ly ngắn và trung bình, và cũng có đóng góp nhất định vào việc tạo cơ hội ghi bàn.

• **Cụm 2 (Xanh lá cây): Các cầu thủ tấn công cánh kỹ thuật**

- *Vị trí:* Cụm này bao gồm các tiền vệ cánh và tiền đạo cánh.
- *Đặc điểm nổi bật:*
 - * Các cầu thủ trong cụm này nổi bật với khả năng đi bóng lắt léo, tạo đột biến và thực hiện các đường chuyền vào vòng cấm đối phương.
 - * Họ có xu hướng ghi bàn và kiến tạo cơ hội cho đồng đội.
 - * Các chỉ số phòng ngự không phải là điểm mạnh của họ.

• **Cụm 3 (Đỏ): Các tiền vệ phòng ngự mạnh mẽ**

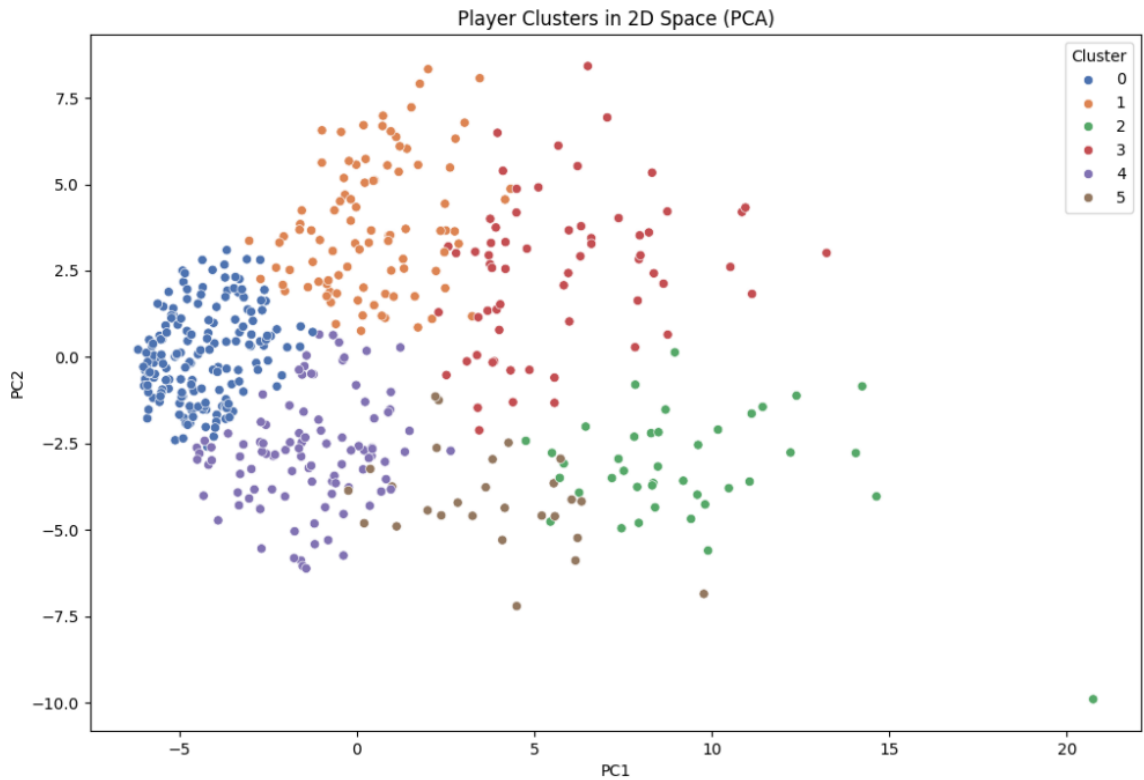
- *Vị trí:* Cụm này tập trung vào các tiền vệ phòng ngự và trung vệ.
- *Đặc điểm nổi bật:*
 - * Các cầu thủ ở cụm này có khả năng phòng ngự mạnh mẽ, với các chỉ số tắc bóng, đánh chặn và tranh chấp bóng cao.
 - * Họ tập trung chủ yếu vào việc thu hồi bóng và bảo vệ hàng thủ.
 - * Khả năng tấn công và kiến tạo của họ thường hạn chế hơn.

• **Cụm 4 (Tím): Các tiền đạo mục tiêu**

- *Vị trí:* Cụm này chủ yếu bao gồm các tiền đạo cắm.
- *Đặc điểm nổi bật:*
 - * Các cầu thủ trong cụm này có xu hướng chơi ở gần khung thành đối phương và tham gia vào các tình huống không chiến.
 - * Họ có khả năng ghi bàn tốt, đặc biệt là từ các pha dứt điểm trong vòng cấm.
 - * Họ có thể không tham gia nhiều vào việc xây dựng lối chơi.

• **Cụm 5 (Nâu): Các thủ môn**

- *Vị trí:* Cụm này bao gồm các thủ môn.
- *Đặc điểm nổi bật:*
 - * Các chỉ số quan trọng nhất của họ liên quan đến khả năng cản phá, bắt bóng và phản xạ.
 - * Các chỉ số tấn công và chuyền bóng của họ rất thấp.



Hình 3.2: Hình ảnh biểu đồ phân tán 2D

Các tập tin CSV `cluster_0_players.csv`, `cluster_1_players.csv`, `cluster_2_players.csv`, `cluster_3_players.csv`, `cluster_4_players.csv` và `cluster_5_players.csv` chứa dữ liệu chi tiết về các cụm tương ứng.

3.3 Kết luận

Thuật toán K-means đã được áp dụng thành công để phân loại các cầu thủ thành 6 nhóm dựa trên các chỉ số thống kê. Phương pháp Elbow cung cấp một cách tiếp cận dựa trên dữ liệu để chọn số lượng cụm tối ưu. PCA giúp đơn giản hóa việc trực quan hóa các cụm trong không gian 2D. Phân tích cụm chi tiết làm sáng tỏ các đặc điểm thống kê quan trọng phân biệt các nhóm cầu thủ khác nhau.

Chương 4

Ước tính giá trị cầu thủ

4.1 Giới thiệu và cách thu thập dữ liệu

4.1.1 Giới thiệu

Thị trường chuyển nhượng cầu thủ bóng đá là một lĩnh vực kinh tế sôi động, nơi giá trị của các cầu thủ đóng vai trò quan trọng trong các giao dịch. Việc ước tính chính xác giá trị này mang lại lợi ích to lớn cho các câu lạc bộ trong việc định giá tài sản, đàm phán hợp đồng, cũng như cho người đại diện và các bên liên quan khác. Chương này tập trung vào việc xây dựng và đánh giá một phương pháp ước tính giá trị chuyển nhượng của cầu thủ dựa trên dữ liệu thu thập được từ trang web <https://www.footballtransfers.com>. Mục tiêu chính là cung cấp một công cụ hỗ trợ định lượng để đưa ra các quyết định sáng suốt hơn trong thị trường chuyển nhượng, thay vì chỉ dựa vào cảm tính hoặc kinh nghiệm cá nhân. Phương pháp này kết hợp các yếu tố thống kê về hiệu suất cầu thủ, thông tin cá nhân và yếu tố câu lạc bộ để tạo ra một mô hình dự đoán toàn diện.

4.1.2 Thu thập dữ liệu

Dữ liệu giá trị chuyển nhượng của cầu thủ cho mùa giải 2024-2025 được thu thập từ trang web <https://www.footballtransfers.com> bằng kỹ thuật web scraping. Quá trình thu thập dữ liệu được tự động hóa để thu thập thông tin từ nhiều trang và cầu thủ. Để đảm bảo tính đại diện và độ tin cậy của dữ liệu, chỉ những cầu thủ có thời gian thi đấu lớn hơn 900 phút được đưa vào phân tích. Điều này giúp loại bỏ những cầu thủ ít có đóng góp thực tế và tập trung vào những người chơi thường xuyên ra sân.

4.2 Phương pháp ước tính giá trị cầu thủ

4.2.1 Lựa chọn đặc trưng (Feature Selection)

Các đặc trưng được lựa chọn để ước tính giá trị cầu thủ bao gồm:

- Thông tin cá nhân: Tuổi, vị trí thi đấu
- Thống kê thi đấu: Số phút thi đấu, số bàn thắng, kiến tạo, v.v. (từ file results.csv)
 - File results.csv chứa dữ liệu thống kê chi tiết về màn trình diễn của cầu thủ trong mùa giải, được thu thập từ bài toán trước đó (Bài 1). Các chỉ số này phản ánh khả năng và đóng góp của cầu thủ trên sân.

- Thông tin câu lạc bộ: Câu lạc bộ hiện tại

Home

results.csv

Menu

Format Painter

Format Painter

Format Painter

Copy formatting from one place and apply it to another.

Double-click this button to apply the same formatting to multiple places in the document.

Abdul Fatahi eng GHA

Adam Arsmi eng ENG

Adam Lallier eng ENG

Adam Smith eng ENG

Adam Webster eng ENG

Adam Whar eng ENG

Adama Trac es ESP

Albert Granri dk DEN

Alejandro G ar ARG

Alex Iwobi ng NGA

Alex McCarthy eng ENG

Alex Palmer eng ENG

Alex Scott eng ENG

Alexander Is se SWE

Alexis Mac ar ARG

Ali Al Hamai iq IRQ

Alisson br BRA

Alphonse Ar fr FRA

Altay Bayin tr TUR

Amad Diallo ci CIV

Amadou On be BEL

Andreas Per br BRA

Andreas Rost ert GDR

Home

Insert

Page Layout

Formulas

Data

Review

View

Tools

Smart Toolbox

Share

Settings

Calibri

11

General

Conditional Formatting

Data Processing

Smart Toolbox

Settings

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Team	Position	Age	Playing Time	Playing Time	Performance	Performance	Performance	Performance	Expected: e	Expected: e	Progression	Progression	Progression	Per 90 min	Per 90 min	Per 90 min	Per 90 min	Per 90 min
2	West Ham	DF	35-146	15	8	676	0	0	2	0	0.1	1.1	4	28	2	0	0	0.01	
3	Southampton	GK	26-361	27	27	2430	0	0	2	0	0	0	0	0	0	0	0	0	0
4	West Ham	DF	27-165	33	32	2884	2	3	1	0	1.2	3.2	101	132	150	0.06	0.09	0.04	
5	Everton	MF	32-129	30	29	2425	3	1	5	1	3.9	2.3	40	78	91	0.11	0.04	0.14	
6	Manchester	DF	21-070	6	6	503	0	0	1	0	0	0	1	25	2	0	0	0	
7	Leicester	CF	21-063	11	6	579	0	2	0	0	0.4	1.6	42	17	60	0	0.31	0.06	
8	Adam Arsmi eng ENG	Southampton	FW,MF	28-089	20	15	1248	2	2	4	0	3.3	1.2	25	21	79	0.14	0.14	0.24
9	Adam Lallier eng ENG	Southampton	MF	37-000	14	5	361	0	2	4	0	0.2	0.9	6	24	10	0	0.5	0.04
10	Adam Smith eng ENG	Bournemouth	DF	34-011	22	17	1409	0	0	6	0	0.7	0.3	12	40	31	0	0	0.04
11	Adam Webster eng ENG	Brighton	DF	30-126	11	8	617	0	0	0	0	0	0.5	7	40	2	0	0	0
12	Adam Whar eng ENG	Crystal Pala	MF	20-342	20	16	1318	0	2	2	0	0.4	3	14	107	11	0	0.14	0.03
13	Adama Trac es ESP	Fulham	FW,MF	29-105	33	16	1592	2	6	3	0	3.9	4.7	89	61	145	0.11	0.34	0.22
14	Albert Granri dk DEN	Southampton	FW,MF	23-352	4	2	143	0	0	0	0	0	0	1	1	3	0	0	0.07
15	Alejandro G ar ARG	Manchester	FW,FW	20-313	34	23	2146	6	2	3	0	7.2	4.5	139	56	281	0.25	0.08	0.3
16	Alex Iwobi ng NGA	Fulham	FW,MF	29-007	35	33	2796	9	6	1	0	4.6	6.9	135	199	224	0.29	0.19	0.15
17	Alex McCarthy eng ENG	Southampton	GK	35-158	5	5	450	0	0	0	0	0	0	0	0	0	0	0	0
18	Alex Palmer eng ENG	Ipswich Tow	GK	28-273	11	11	990	0	0	2	0	0	0	0	1	0	0	0	0
19	Alex Scott eng ENG	Bournemouth	MF	21-262	18	7	709	0	0	2	0	0.7	0.8	15	49	33	0	0	0.08
20	Alexander Is se SWE	Newcastle	FW	25-231	32	32	2577	23	6	1	0	19.9	4.3	81	80	200	0.8	0.21	0.69
21	Alexis Mac ar ARG	Liverpool	MF	26-137	34	30	2575	5	5	6	0	2.8	4.6	34	174	79	0.17	0.17	0.1
22	Ali Al Hamai iq IRQ	Ipswich Tow	FW	23-070	11	0	134	0	0	3	0	0.4	0.1	4	3	14	0	0	0.24
23	Alisson br BRA	Liverpool	GK	32-220	25	25	2238	0	0	0	0	0	0	0	0	0	0	0	0
24	Alphonse Ar fr FRA	West Ham	GK	32-072	24	23	2080	0	0	0	0	0	0	0	0	0	0	0	0
25	Altay Bayin tr TUR	Manchester	GK	27-026	2	2	180	0	0	0	0	0	0	0	0	0	0	0	0
26	Amad Diallo ci CIV	Manchester	FW,MF	22-303	23	17	1639	7	6	3	0	4.2	4	93	55	163	0.38	0.33	0.23
27	Amadou On be BEL	Aston Villa	MF	23-267	23	17	1378	3	0	4	0	2.1	0.3	21	60	14	0.2	0	0.14
28	Andreas Per br BRA	Fulham	MF	29-129	31	23	1879	2	4	7	0	3.5	4.5	27	99	78	0.1	0.19	0.17
29	Andreas Rost ert GDR	Liverpool	DF	21-190	21	22	2388	0	0	2	1	1	4.1	58	165	95	0	0	0.04

Hình 4.1: Hình ảnh file results.csv

Việc lựa chọn các đặc trưng này dựa trên giả định rằng chúng có ảnh hưởng đáng kể đến giá trị của một cầu thủ trên thị trường chuyển nhượng. Ví dụ, tuổi tác và vị trí thi đấu thường ảnh hưởng đến tiềm năng và thời gian thi đấu còn lại của một cầu thủ, trong khi thống kê thi đấu phản ánh phong độ và hiệu suất thực tế.

4.2.2 Lựa chọn mô hình

Mô hình Gradient Boosting Regression (GBR) được sử dụng để ước tính giá trị cầu thủ. GBR là một thuật toán mạnh mẽ có khả năng nắm bắt các mối quan hệ phi tuyến phức tạp giữa các đặc trưng và biến mục tiêu (giá trị chuyển nhượng). So với các phương pháp đơn giản hơn như hồi quy tuyến tính, GBR có khả năng mô hình hóa các tương tác phức tạp giữa các biến và đưa ra dự đoán chính xác hơn.

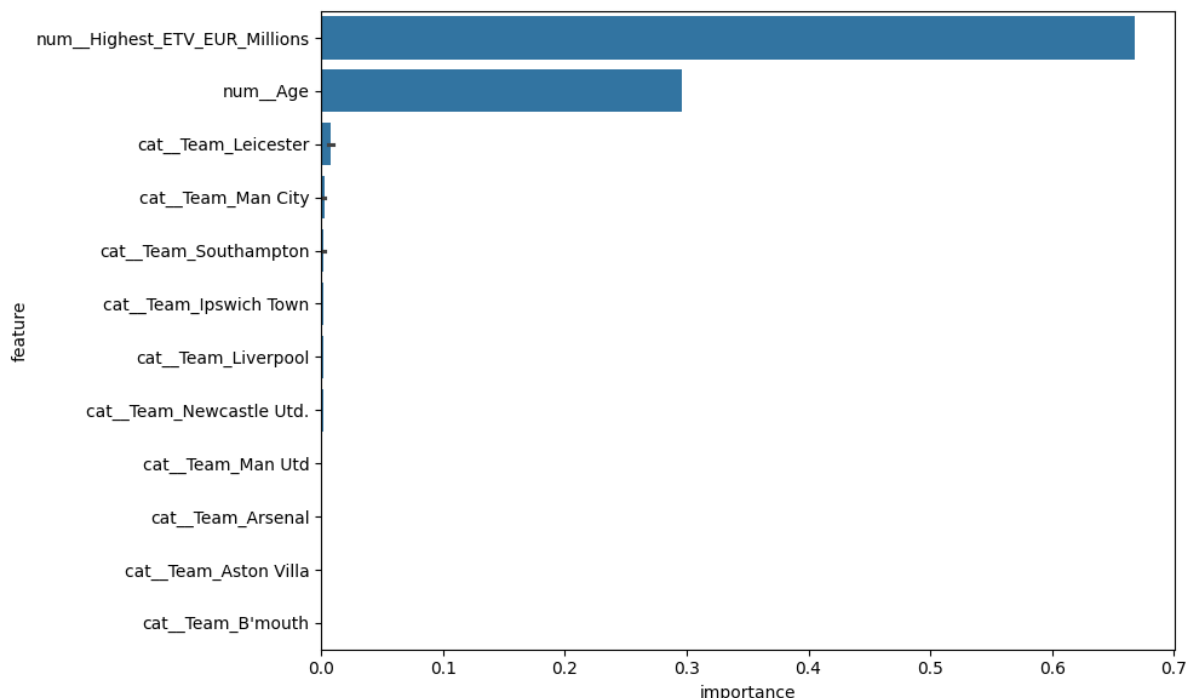
4.2.3 Phân tích hình ảnh trực quan hóa về hiệu suất của mô hình

Để trực quan hóa và hiểu rõ hơn về hiệu suất của mô hình, ba hình ảnh sau đây được sử dụng:

- **feature_importance.png**: Biểu đồ thể hiện mức độ quan trọng của các đặc trưng đầu vào trong việc dự đoán giá trị chuyển nhượng.
- **pred_vs_actual.png**: Biểu đồ so sánh giữa giá trị chuyển nhượng dự đoán và giá trị thực tế.
- **residuals.png**: Biểu đồ phần dư (hiệu giữa giá trị thực tế và dự đoán) để đánh giá tính phân phối và xu hướng của sai số.

Biểu đồ mức độ quan trọng của các đặc trưng đầu vào (feature_importance.png)

Biểu đồ này cho thấy các đặc trưng có vai trò quan trọng nhất trong việc dự đoán giá trị chuyển nhượng. Các đặc trưng có mức độ quan trọng cao hơn có ảnh hưởng lớn hơn đến kết quả dự đoán của mô hình.



Hình 4.2: Hình ảnh của biểu đồ feature_importance

Nhận xét:

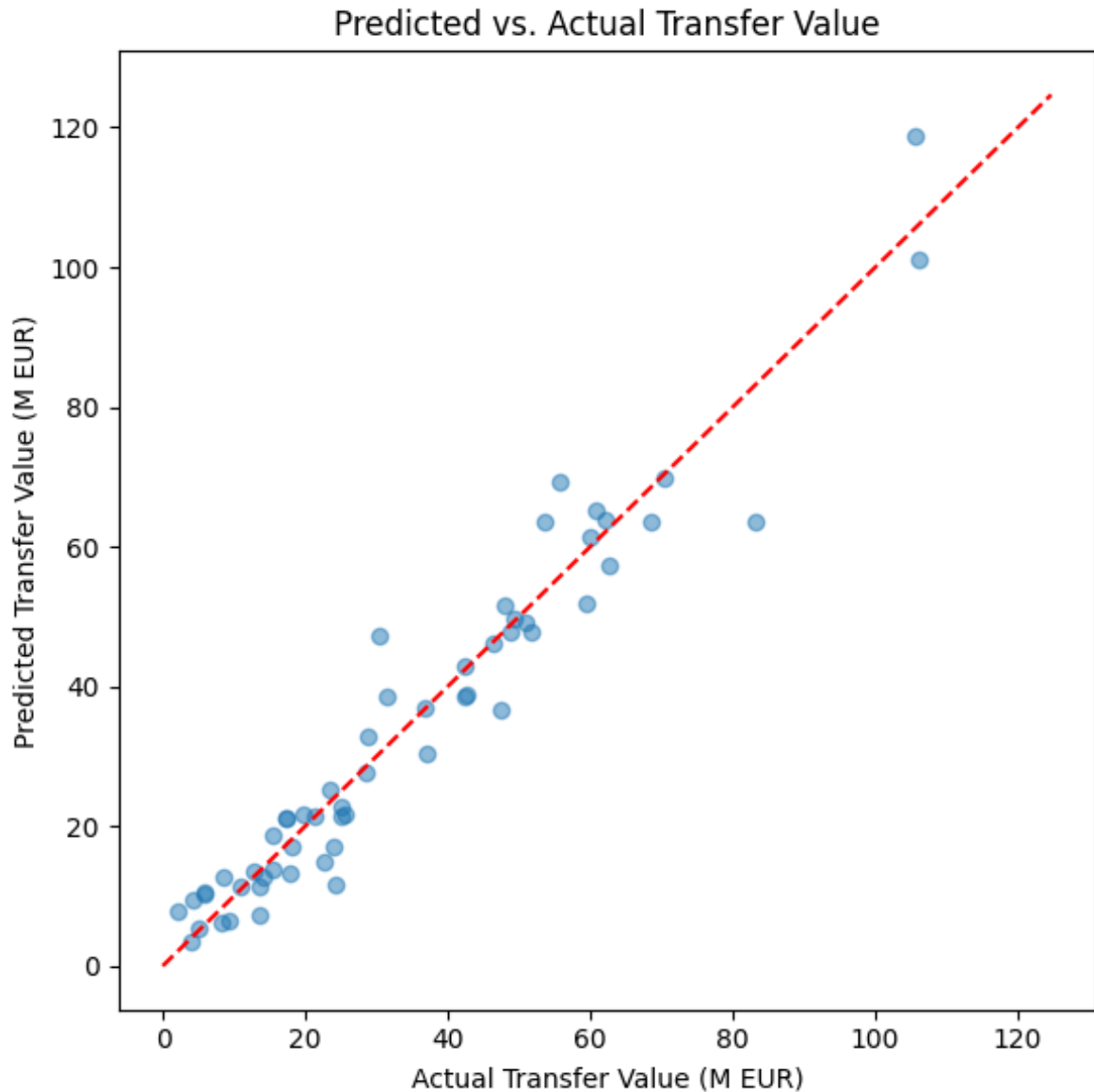
- Cần xác định tên các đặc trưng cụ thể từ biểu đồ (ví dụ: "Age", "Position", "Goals", "Assists", v.v.).
- Phân tích thứ tự và sự chênh lệch về mức độ quan trọng giữa các đặc trưng.
- Giải thích ý nghĩa của mức độ quan trọng của từng đặc trưng trong ngữ cảnh bóng đá (ví dụ: "Số bàn thắng" có thể quan trọng vì phản ánh khả năng ghi bàn và sức hút thương mại của cầu thủ).

Biểu đồ so sánh giá trị dự đoán và thực tế (pred_vs_actual.png)

Biểu đồ này hiển thị mối quan hệ giữa giá trị chuyển nhượng dự đoán bởi mô hình và giá trị chuyển nhượng thực tế. Các điểm dữ liệu gần đường chéo ($y = x$) cho thấy mô hình dự đoán chính xác.

Nhận xét:

- Đánh giá mức độ phân tán của các điểm dữ liệu xung quanh đường chéo.
- Nhận xét về xu hướng của mô hình (ví dụ: có xu hướng đánh giá cao hay thấp giá trị của một số cầu thủ?).
- Xác định các điểm dữ liệu ngoại lệ (nếu có) và suy đoán về nguyên nhân (ví dụ: cầu thủ có giá trị cao bất thường do yếu tố thương mại).



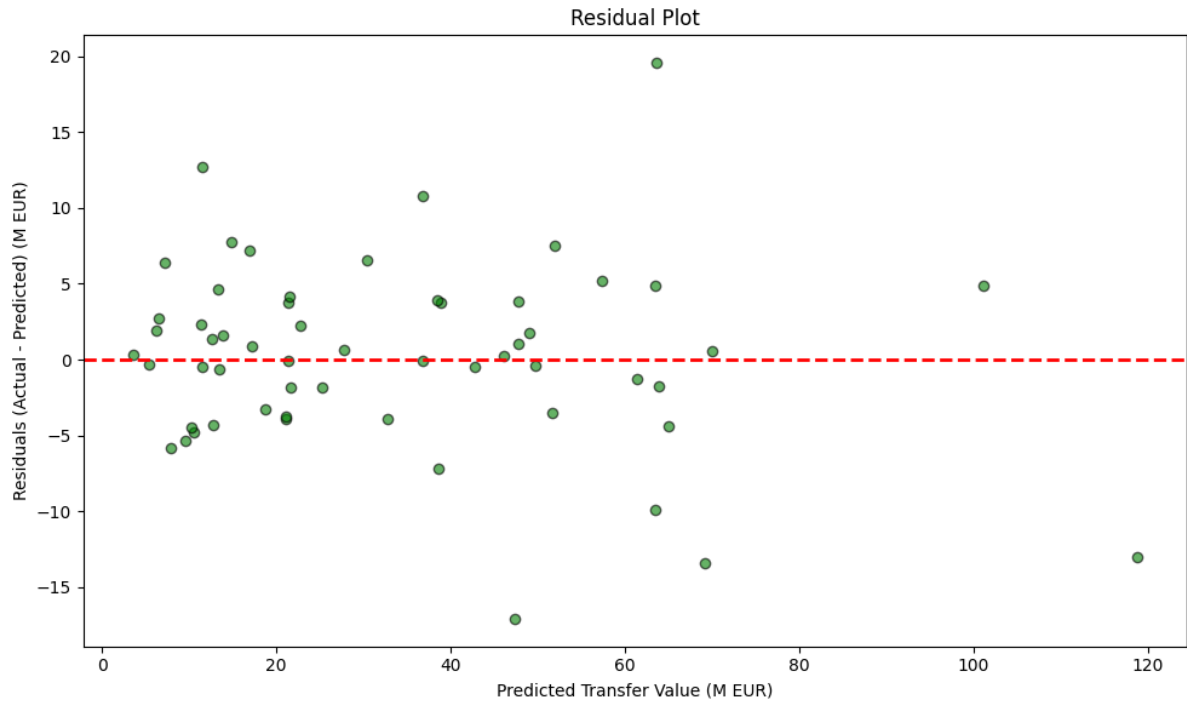
Hình 4.3: Hình ảnh của biểu đồ `pred_vs_actual`

Biểu đồ phần dư (`residuals.png`)

Biểu đồ này hiển thị phân phối của phần dư, tức là sai số giữa giá trị thực tế và giá trị dự đoán. Một mô hình tốt có phần dư phân phối ngẫu nhiên xung quanh 0, không có xu hướng rõ rệt.

Nhận xét:

- Đánh giá tính đối xứng và phân phối của phần dư (ví dụ: có dạng chuông không?).
- Kiểm tra xem phần dư có xu hướng tăng hoặc giảm theo giá trị dự đoán không.
- Xác định các mẫu hình trong phần dư (nếu có) và suy đoán về nguyên nhân (ví dụ: mô hình có thể bỏ qua một yếu tố quan trọng nào đó).



Hình 4.4: Hình ảnh của biểu đồ residuals

4.2.4 Huấn luyện và đánh giá mô hình

- Dữ liệu được chia thành tập huấn luyện (80%) và tập kiểm tra (20%) để đánh giá khả năng tổng quát hóa của mô hình.
- Mô hình GBR được huấn luyện trên tập huấn luyện với kỹ thuật cross-validation (5-fold) và RandomizedSearchCV để tối ưu hóa các tham số. Cross-validation giúp đánh giá mô hình trên nhiều phần dữ liệu khác nhau, trong khi RandomizedSearchCV tìm kiếm các tham số tốt nhất cho mô hình.
- Hiệu suất của mô hình được đánh giá trên tập kiểm tra bằng các chỉ số R2, RMSE và MAE.

4.2.5 Kết quả

```
[Running] python -u "d:\Python\BTL\SourceCode\Ex4\estimate.py"
Fitting 5 folds for each of 30 candidates, totalling 150 fits
R2: 0.937, RMSE: 6.11, MAE: 4.43
```

Hình 4.5: Hình ảnh kết quả đánh giá mô hình

Kết quả huấn luyện và đánh giá mô hình cho thấy:

- Fitting 5 folds for each of 30 candidates, totalling 150 fits: Quá trình tối ưu hóa mô hình đã thử nghiệm 30 bộ tham số khác nhau trên 5 phần dữ liệu (5-fold cross-validation), dẫn đến tổng cộng 150 lần huấn luyện và đánh giá mô hình. Kỹ thuật cross-validation giúp đảm bảo tính ổn định và đáng tin cậy của kết quả, tránh tình trạng overfitting (mô hình chỉ hoạt động tốt trên dữ liệu huấn luyện).

- R2: 0.937: Hệ số xác định (R2) là một chỉ số thống kê đo lường mức độ phù hợp của mô hình với dữ liệu. Giá trị R2 nằm trong khoảng từ 0 đến 1.
 - $R2 = 1$: Mô hình dự đoán hoàn hảo, mọi biến động của biến mục tiêu (giá trị chuyển nhượng) được giải thích bởi các đặc trưng đầu vào.
 - $R2 = 0$: Mô hình không có khả năng dự đoán, không giải thích được bất kỳ sự biến động nào của biến mục tiêu.
 - $R2 = 0.937$: Mô hình của chúng ta giải thích được 93.7% sự biến động của giá trị chuyển nhượng, cho thấy khả năng khớp dữ liệu rất tốt. Điều này có nghĩa là mô hình có thể dự đoán giá trị chuyển nhượng của cầu thủ một cách chính xác dựa trên các đặc trưng đã chọn.
- RMSE: 6.11: Root Mean Squared Error (RMSE) là một chỉ số đo lường sai số trung bình giữa giá trị thực tế và giá trị dự đoán. RMSE càng nhỏ, mô hình càng chính xác.
 - $RMSE = 6.11$: Sai số trung bình giữa giá trị thực tế và giá trị dự đoán là 6.11 triệu EUR. Đây là một mức sai số tương đối thấp so với phạm vi giá trị chuyển nhượng thực tế, cho thấy mô hình có độ chính xác cao.
- MAE: 4.43: Mean Absolute Error (MAE) cũng là một chỉ số đo lường sai số, nhưng khác với RMSE, MAE tính trung bình của các sai số tuyệt đối (không bình phương). MAE ít nhạy cảm hơn với các giá trị ngoại lệ so với RMSE.
 - $MAE = 4.43$: Sai số trung bình tuyệt đối giữa giá trị thực tế và giá trị dự đoán là 4.43 triệu EUR. Giá trị MAE thấp càng khẳng định độ chính xác của mô hình.

4.2.6 Hạn chế

Mô hình ước tính giá trị cầu thủ đạt được hiệu suất ấn tượng, tuy nhiên, vẫn tồn tại một số hạn chế:

- Thiếu sót các yếu tố định tính: Mô hình hiện tại chủ yếu dựa vào các yếu tố định lượng (thống kê, tuổi tác), bỏ qua các yếu tố định tính quan trọng như danh tiếng của cầu thủ, kinh nghiệm thi đấu quốc tế, khả năng lãnh đạo, v.v...
- Tính tổng quát hóa: Mô hình được huấn luyện trên dữ liệu của một giải đấu cụ thể (Premier League), do đó, khả năng áp dụng cho các giải đấu khác có thể bị hạn chế do sự khác biệt về môi trường bóng đá và thị trường chuyển nhượng.
- Sai số dự đoán: Mặc dù các chỉ số đánh giá là tốt, vẫn có một mức độ sai số nhất định trong dự đoán. Điều này có thể do sự phức tạp của thị trường chuyển nhượng và ảnh hưởng của các yếu tố ngẫu nhiên.

4.3 Kết luận và kiến nghị

Mô hình GBR là một công cụ hữu ích để ước tính giá trị chuyển nhượng cầu thủ, vượt trội so với các phương pháp đơn giản hơn như hồi quy tuyến tính nhờ khả năng xử lý các mối quan hệ phi tuyến. Tuy nhiên, để nâng cao tính chính xác và ứng dụng thực tế của mô hình, cần xem xét các kiến nghị sau:

- Bổ sung yếu tố định tính: Nghiên cứu các phương pháp định lượng hóa các yếu tố định tính (ví dụ: sử dụng sentiment analysis để đánh giá danh tiếng cầu thủ từ tin tức và mạng xã hội).
- Mở rộng phạm vi áp dụng: Thu thập dữ liệu từ nhiều giải đấu khác nhau và xây dựng các mô hình riêng biệt hoặc mô hình đa nhiệm để tăng tính tổng quát hóa.
- Cải tiến mô hình: Thử nghiệm các thuật toán học máy khác (ví dụ: Random Forest, Neural Networks) và kết hợp chúng để tạo ra mô hình ensemble mạnh mẽ hơn.

Tài liệu tham khảo

- [1] McKinney, W. (2017). *Python for Data Analysis*. O'Reilly Media. Truy cập ngày 6 tháng 5 năm 2025, từ <https://wesmckinney.com/pages/book.html>
- [2] Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras TensorFlow*. O'Reilly Media.
- [3] Selenium Developers. (2024). *Selenium Documentation*. Truy cập ngày 6 tháng 5 năm 2025, từ <https://www.selenium.dev/documentation/>
- [4] BeautifulSoup Developers. (2024). *Beautiful Soup Documentation*. Truy cập ngày 6 tháng 5 năm 2025, từ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [5] Scikit-learn Developers. (2024). *Scikit-learn User Guide*. Truy cập ngày 6 tháng 5 năm 2025, từ https://scikit-learn.org/stable/user_guide.html
- [6] Healy, K. (2019). *Data Visualization with Python*. O'Reilly Media.